Sistema de recomanació de tractaments mèdics per a pacients amb malalties respiratòries

Aleix Trasserra

Director: David Juan i Romero

. Ponent: Javier Béjar (Departament de Ciències de la Computació)

Especialitat: Computació

Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya

Abril de 2018

Resum

El projecte consisteix a realitzar una aplicació que doni suport a pacients amb malalties respiratòries combinant tècniques de Machine Learning i Gamification. Tracta d'una prova de concepte en forma d'aplicació d'escriptori que fa prediccions sobre variables clíniques de pacients.

Resumen

El proyecto consiste a realizar una aplicación que de soporte a pacientes que padecen enfermedades respiratorias combinando técnicas de Machine Learning i Gamification. Trata de una prueba de concepto en formato aplicación de escritorio que hace predicciones sobre variables clínicas de pacientes.

Abstract

The project consists in making an application that supports patients with respiratory diseases by combining Machine Learning and Gamification techniques. It deals with a concept test in the form of a desktop application that consists of making predictions on clinical variables of patients.

Índex

Índex de figures

Índex de taules

1 Contextualització i abast del projecte

1.1 Contextualització

En el camp de l'atenció mèdica es generen grans quantitats de dades a diari. Els centres mèdics en general(Hospitals, Centres d'atenció primària, etc.) prenen mesures de tot tipus als pacients: nivell d'oxigen en sang, pressió arterial i un llarguíssim etcètera. Per a observar de forma més palpable la magnitud del volum de dades, segons la llei [2] els centres mèdics han de conservar la documentació clínica 5 anys després de cada alta mèdica a l'estat Espanyol. Aquest volum de dades genera uns costos de manteniment i emmagatzematge. No obstant, aquestes dades són alhora un gran potencial d'informació que pot ser aprofitat per millorar la qualitat dels serveis mèdics.
En l'article , s'exposa la necessitat i la oportunitat d'aprofitar el coneixement científic i tècnic per explotar la informació generada i ajudar a la recerca en el camp de la medicina.
Gràcies a l'avenç de tecnologies com el Big Data o el Machine Learning , l'explotació de dades i l'ús intel·ligent d'aquesta explotació és ara més possible que mai tal com s'exposa en l'article [3] on a més també proposa possibles subcamps de la medicina on les tecnologies esmentades poden aportar solucions a problemes existents.
Un altre objectiu en l'atenció mèdica és donar una atenció personalitzada als pacients. El gran nombre de pacients que han d'atendre els professionals fa que es generin cues i dificulta la possibilitat que el pacient tingui un seguiment temporal adequat. L'auge tan de les aplicacions mòbils/web com de les Intranets fan possible que els pacients pugin tenir un feedback fluid del seu estat i també un seguiment en temps real de l'evolució dels tractaments que se li estan realitzant.
Una possible millora en la qualitat de l'atenció mèdica és incorporar tècniques de Ludificació (Gamification en anglès) a aquestes aplicacions. Segons [4] la ludificació és l'ús dels elements i de la mecànica del joc en contextos aliens a aquest amb l'objectiu d'orientar el comportament de les persones i aconseguir determinades fites. En el cas de la medicina, es pot utilitzar per donar motivació al pacient per seguir el tractament i les recomanacions del prescriptor.

1.2 Objectiu del treball

Un cop introduït el context, en aquesta secció s'exposarà en que consisteix el treball.
El treball consistirà en desenvolupar una aplicació destinada a pacients que pateixin malalties respiratòries. Aquesta aplicació consistirà a desenvolupar un sistema basat en Machine Learning que sigui capaç de predir valors de diferents variables indicatives en el

tractament que estigui seguint el pacient (per exemple el nivell d'oxigen en sang). A més de veure els resultats d'aquestes prediccions, també hi haurà un apartat on el pacient podrà veure diferents fites que vindran marcades en funció de les prediccions, aplicant així tècniques de Ludificació. Opcionalment, l'aplicació permetrà al pacient introduir a l'aplicació observacions personals en format text que permetran al sistema recopilar més informació i aprendre sobre aquesta mitjançant tècniques de topic extraction i sentiment analysis.

1.3 Actors implicats

Els principals actors implicats en el projecte són els següents:

- **Client**: el client és l'empresa que ofereix tractaments i serveis a persones amb malalties respiratòries. També donarà feedback sobre l'evolució del projecte i serà l'encarregat de concretar certs aspectes de l'aplicació. A més, cedirà i explicarà les dades necessàries per al desenvolupador per tal de poder treballar i entendre millor el model de dades.
- **Desenvolupador**: l'autor d'aquesta memòria serà l'únic encarregat de desenvolupar l'aplicació i de documentar els resultats.
- **Beneficiaris**: l'empresa per la qual treballa l'autor d'aquesta memòria serà el principal beneficiat del treball ja que tindrà la possibilitat d'usar el coneixement del desenvolupador i els resultats d'aquest treball.
- **Usuaris**: els possibles usuaris de l'aplicació són els clients de l'empresa que es dedica a oferir tractaments i serveis a persones amb problemes respiratoris.
- **Director del projecte**: Javier Béjar serà l'encarregat de dirigir el projecte, guiant al desenvolupador i donant-li suport tan tècnic com de gestió del projecte.
- **Cap de projectes de l'empresa**: David Juan serà l'encarregat de donar suport i ajuda durant el dia a dia mentre duri el desenvolupament dins l'empresa i també l'encarregat de fer d'intermediari entre el client i el desenvolupador.

1.4 Abast del projecte

El projecte es pretén que sigui una prova de concepte. No es pretén que l'aplicació sigui un producte completament acabat i capaç de sortir a producció. El client vol veure si el Machine Learning té cabuda en el seu "stack" tecnològic i de quina forma pot millorar els seus serveis als clients. El client també vol una primera presa de contacte amb les tècniques de Ludificació i el grau satisfacció que aporta aquesta nova forma d'afrontar els tractaments mèdics envers la forma en que ho fan fins ara.
Pel que fa a la qualitat de les prediccions, el client desitja que aquestes proporcionin avantatges als prescriptors. No obstant, degut a que el projecte és una prova de concepte,

no s'exigeix arriba a cap tant per cent d'encert en les prediccions ni una robustesa extrema del model.
Conèixer les dades i saber adaptar-les al context de la Intel·ligència artificial i el Big Data és un aspecte que està a l'abast del projecte i és un objectiu més prioritari que no pas aconseguir models predictius d'encert elevat. El client vol tenir una base sòlida i una bona comprensió de les dades per a poder aplicar aquestes tècniques en projectes futurs relacionats.

1.5 Estat de l'art

El Machine Learning és una de les tecnologies en la qual el món de la medicina està dedicant més esforços en aplicar ja que permet passar a aplicar mètodes proactius.
Centrant-nos en la prescripció personalitzada de tractaments, l'empresa IBM ha desenvolupat un producte anomenat IBM Watson Health que recomana les possibles millors opcions per a tractaments en el camp de l'oncologia i ajuda als pacients a entendre cadascuna d'aquestes opcions. En la pàgina [5] podem veure la descripció detallada del producte.
Pel que fa al camp de les malalties respiratòries, estudiantes del Coimbatore College for Women a la Índia [6] han desenvolupat un estudi on exposen estratègies per detectar malalties pulmonàries basat en Machine Learning. En l'estudi obtenen, en el millor dels casos, un 92.34% d'encert en les prediccions i conclouen que els resultats obtinguts indiques que el predictor pot ser útil per als prescriptors.
Una altre estudi relacionat és el que podem llegir a [5] on utilitzen Deep Learning per predir si un pacient haurà de ser reingressat o no en un hospital. Més en concret, la intenció es predir si hi haurà un canvi d'estat en la hospitalització del pacient en un futur (passar de ingressat a donat d'alta, si es mantindrà ingressat, etc.). L'estudi és interessant ja que explica molt clarament el model de dades utilitzat i el mètode de preprocessat. El model és molt similar al que ens cedeix el client i l'estudi pot servir com a possible referència de cara a treballar amb les dades en el treball.
A [6] es descriu un mètode que, utilitzant tècniques de Unsupervised Learning, són capaços d'identificar possibles anomalies en la malaltia del pacient i també per predir l'evolució del tractament. A l'estudi es proposa que les tècniques presentades poden permetre als prescriptors entendre millor l'evolució dels tractaments podent així optimitzarlos i dissenyar noves guies per a aquests.
Per altra banda, les tècniques de Ludificació s'han aplicat en conceptes de medicina relacionats amb l'activitat física i el problema de la obesitat. Existeixen diverses aplicacions que promouen l'activitat física mitjançant petits reptes diaris o jocs diversos. No obstant, aquestes tècniques també s'estan començant a aplicar en la resta de camps de la medicina. Un bon exemple és la app mySugr que proposa reptes diaris a pacients amb diabetis a més d'oferir comentaris motivadors per incitar al pacient a seguir i millorar el tractament.

1.6 Metodologia

Degut a que aquest projecte només comptarà amb un desenvolupador i el client no és expert en la matèria, s'utilitzarà una metodologia Agile ja que aquesta es centra en el feedback directe i continuat entre desenvolupador i client. Es faran reunions setmanals i durant aquestes es fixaran els objectius per a la següent.
A gran escala, el projecte tindrà les següents iteracions:

1. Adaptació de les dades al Big Data.
2. Creació del model/models i avaluació.
3. Creació d'una App que utilitzi el model.
4. Introducció de tècniques de Ludificació a l'aplicació.

1.7 Eines de monitorització i validació

L'eina principal de monitorització i seguiment de l'evolució del projecte serà TFS(Team Foundation Server) [7] de la companyia Microsoft. TFS proporciona un conjunt d'eines de desenvolupament de software que permeten compartir codi, control de versions, eines per a aplicar metodologies. Aquesta eina serà utilitzada principalment pel desenvolupador. El client i el cap de projectes tindran accés al repositori per tal de poder fer un seguiment i informar al desenvolupador de possibles incidències si s'escau.
Per validar que el projecte evolucioni correctament s'utilitzarà l'eina de TFS que permet reportar errors a més de les reunions periòdiques amb el client. El director del projecte tindrà el paper de verificar que la qualitat del projecte és la que correspon a un treball de final de grau. Finalment, el Cap de projectes de l'empresa verificarà que la relació entre client i desenvolupador sigui fluida i també tindrà poder de decisió en varies de les decisions que es prenguin.

1.7.1 Possibles obstacles

Els possibles obstacles que poden aparèixer durant la realització del projecte són els següents:

- **Qualitat de les dades**: en funció de la qualitat de les dades i de l'estructura d'aquestes serà més fàcil o més complicat construir els predictors. Si l'estructura és complicada, s'hauran de dedicar més esforços a la comprensió. Per altra banda, si la qualitat de les dades és dolenta s'haurà d'augmentar la inversió del temps en el preprocessat.
- **Planificació**: com en tot projecte, una mala planificació pot comportar la no finalització d'aquest o que la qualitat se'n vegi ressentida. La naturalesa de les dades pot ser un dels punts crítics en la planificació. És important doncs una bona planificació.

- **Poder computacional**: malgrat que a l'empresa es disposa de bon equipament informàtic, la naturalesa del problema comporta que per a obtenir bons resultats cal processar moltes dades i això pot portar problemes si no es fa una bona tria de les dades o no s'apliquen els mètodes més òptims. Tot i així és possible que el temps computacional segueixi sent un problema i no quedarà més remei que reduir el conjunt de dades per tal de poder abordar el problema en un temps raonable.
- **Selecció dels algorismes adequats per a la creació dels models**: un dels principals punts que tota solució amb una aproximació utilitzant Machine Learning és el de triar adientment els algorismes de creació i parametritzar-los correctament. Una mala tria en general pot comportar una pitjor qualitat dels predictors i de l'aplicació en general.
- **Ludificació**: el desenvolupador no té pràcticament coneixements sobre tècniques de Ludificació. Una bona comunicació amb el client i la pertinent recerca sobre el tema serà cabdal per aplicar correctament aquestes tècniques en l'aplicació.

2 Planificació

En aquesta secció s'explicaran les tasques que es duran a terme durant el projecte així com les possibles alternatives en cas de complicacions o alteracions.
El projecte començarà la setmana de l'11 de setembre de 2017 i està previst que s'acabi la setmana del 15 de gener. És a dir, la durada estimada és 4 mesos.

2.1 Descripció de les tasques

El primer pas per realitzar el projecte serà estudiar el model de dades del qual disposem. Es disposa d'una base de dades relacional. La tasca consisteix en estudiar el model de dades i entendre'l.
Per a dur-la a terme, el client cedirà una base de dades amb les dades personals dels pacients anònimes per tal de protegir la identitat d'aquests.
Es faran diverses reunions amb el client, que és qui coneix el model, i aquest explicarà el model i resoldrà els dubtes pertinents per part del desenvolupador.
Aquesta tasca no hauria de tenir cap inconvenient que fes canviar el pla d'acció, més enllà de la capacitat de comprensió que pot alterar el temps previst.

2.1.1 Reconeixement de les dades

El primer pas per realitzar el projecte serà estudiar el model de dades del qual disposem. Es disposa d'una base de dades relacional. La tasca consisteix en estudiar el model de dades i entendre'l. Per a dur-la a terme, el client cedirà una base de dades amb les dades personals dels pacients anònimes per tal de protegir la identitat d'aquests.
Es faran diverses reunions amb el client, que és qui coneix el model, i aquest explicarà el model i resoldrà els dubtes pertinents per part del desenvolupador.
Aquesta tasca no hauria de tenir cap inconvenient que fes canviar el pla d'acció, més enllà de la capacitat de comprensió que pot alterar el temps previst.

2.1.2 Anàlisi descriptiva

Un cop entès el model de dades, el següent objectiu és fer una anàlisi descriptiva de les dades per tal de tenir una primera aproximació a la forma d'aquestes. Es mostraran gràfiques de les dades que permetin tenir una concepció visual de la forma d'aquestes. A més també es podran mostrar estadístiques d'aquestes (mitjanes, medianes, quantils, etc.).
Aquesta tasca no hauria de tenir cap inconvenient que fes canviar el pla d'acció, més enllà de la capacitat d'anàlisi del desenvolupador.

2.1.3 Preprocessament

Un dels passos més importants a l'hora de crear un bon model de dades és el preprocessament. L'objectiu és, un cop enteses les dades i tenint ja referències sobre la forma, adaptar-les de la forma més precisa al problema per tal de que el model tingui la capacitat d'aprendre millor la forma.
Els passos que caldrà fer són: eliminar variables insignificants per al problema, eliminar elements (files) que no aportin informació, identificar les variables que descriuen més el problema i eliminar les que el descriuen menys si per temes de computació s'han de reduir, substituir valors nuls o que falten.
Tot i que el problema no és de classificació, ens podem trobar amb unes dades no balancejades, possiblement podria interessar aplicar tècniques de balanceig de dades.

2.1.4 Creació i avaluació del model

Quan les dades estiguin adaptades al problema, cal crear un model predictiu que permeti predir la variable de la forma més precisa possible i amb la màxima robustesa. Es provaran diversos algorismes i la idea és que, sobretot, es treballarà amb xarxes neuronals ja que permeten més flexibilitat. Per a aplicar els algorismes disponibles, es separaran les dades entre dades d'entrenament i de test (es provaran diverses proporcions). Els diferents algorismes seran provats aplicant tècniques com Cross- Validation i es compararà l'encert dels diferents models proposats.
Durant el procés pot passar que amb cap algorisme el model es comporti suficientment bé, llavors es tornarà a revisar el preprocés de les dades i es modificarà.
L'objectiu final d'aquest punt és tenir un model predictiu de les dades suficientment precís i robust que pugui ser utilitzat per l'aplicació

2.1.5 Aprenentatge de Xamarín

Xamarin és un conjunt de llibreries programades amb el llenguatge C# que permeten desenvolupar aplicacions mòbils per a plataformes Android, iOS i Windows Phone amb un únic codi.
L'objectiu d'aquesta part és familiaritzar-se amb aquesta tecnologia per tal de poder desenvolupar l'aplicació de forma eficient.

2.1.6 Desenvolupament de la App

Aquesta part consistirà en desenvolupar un prototip d'aplicació utilitzant la tecnologia Xamarin. L'aplicació consistirà en un front-end on l'usuari podrà veure l'estat dels seus tractaments i tindrà l'opció de seleccionar una data per tal de veure la predicció del futur comportament del seu tractament. El back-end de l'aplicació constirà en un servei que cridarà al model per tal de fer la predicció. També tindrà un servei que consultarà les dades del pacient perquè el front-end les pugui mostrar.
És possible que existeixi algun error en el model o que aquest realment no s'adapti tant

a les dades com es preveia en un moment. Per tant, si passa això s'haurà de tornar a revaluar al model i readaptar-lo.

2.1.7 Introducció d'elements de Ludificació a la App

L'objectiu d'aquesta part serà introduir elements de Ludificació a l'aplicació per tal que el pacient a més de veure les seves prediccions pugui donar-se compte de la magnitud del seu comportament i es conscienciï més dels possibles efectes que li poden ocórrer. Aquests elements seran introduïts amb el suport del client, el qual coneix les característiques de l'usuari de l'aplicació

2.1.8 Testing

Finalment, es realitzaran testos unitaris a l'aplicació per tal de comprovar que totes les funcionalitats es comporten de la forma esperada. Malgrat l'aplicació en sí no es preveu que tingui una complexitat enorme. El fet de tenir un model al darrere el qual també ha de funcionar correctament, pot comportar que també s'hagi de fer tests a aquest i això podria augmentar el temps d'aquesta fase.

2.2 Dedicació

Tasca	**Hores dedicades**
Reconeixement de les dades	40
Anàlisi descriptiva	20
Preprocessament	80
Creació i avaluació del model	80
Aprenentatge de Xamarin	20
Desenvolupament de la App	120
Introducció d'elements de Ludificació a la App	40
Testing	100
Total hores dedicades: 500	

Taula 2.1: Temps(H) per tasca del projecte

2.2.1 Diagrama de Gantt

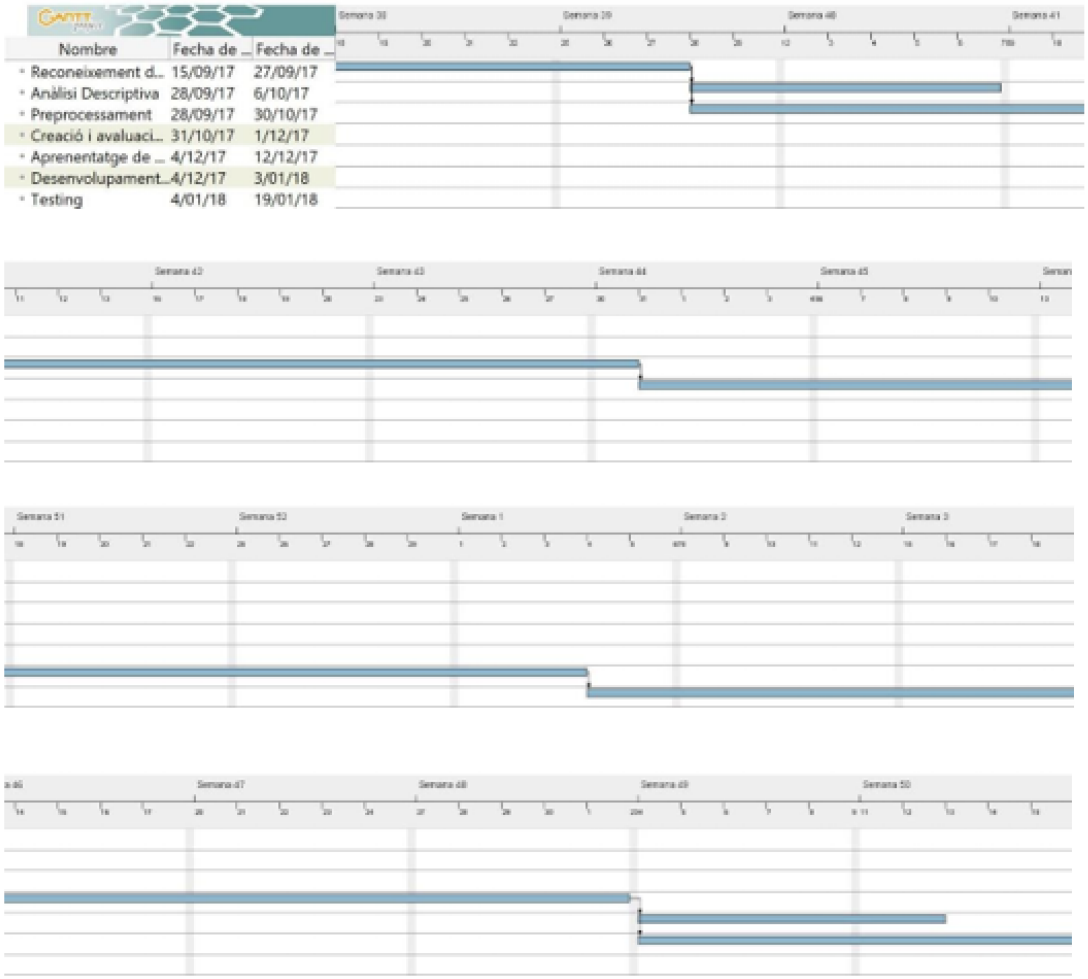

Figura 2.1: Diagrama de Gantt

2.2.2 1 Descripció de les dependències de precedència

A continuació es descriuen les dependències de precedència de les tasques del projecte:

1. Abans de començar la tasca de preprocessament de dades cal haver entès el model de dades i haver-lo analitzat.

2. Abans de començar a crear i avaluar el model cal haver fet un preprocessament de les dades. Això no impedeix però, que mentre s'estigui creant el model calgui fer alguna tasca més de preprocessat.

3. Per començar a crear l'aplicació cal tenir un model de dades el qual pugui ser consumit per la pròpia aplicació.

2.3 Alternatives i pla d'acció

Gràcies a les metodologies Agile que seran aplicades durant l'execució d'aquest projecte, podem revisar i adaptar el projecte "On the Go". De manera que podem fer front a imprevistos de forma ràpida i eficient. Com que l'entrega del projecte final té una data fixada, les alternatives a possibles incidències consistiran en re-calcular el temps de cada tasca del projecte.
Amb una dedicació d'unes 30 hores setmanals al lloc de treball més una revisió extra setmanal d'una hora o dues i 16 setmanes de treball el pla de treball és assumible.
A continuació s'exposen uns quants exemples de possibles incidències que poden ocórrer i el pla d'acció a executar en cas que apareguin:

- Complexitat de les dades: tant si es dona el cas que tenim un volum de dades massa elevat com si el nombre de variables és molt elevat, l'alternativa serà reduir el conjunt de dades i no optimitzar del tot el procés. Si quan s'està avaluant el model es detecta que aquest no es prou bo, es tornarà a revisar el procés.

- Dificultat de trobar un model amb un nivell d'encert adequat: si és dificultós trobar un model amb un encert mínimament acceptable, es simplificarà el model de dades per tal de facilitar aquesta tasca. Aquest fet comportarà afegir un extra de temps a la fase 3 i una reducció en el temps de la resta de tasques.

- Bugs: si apareixen errors molt greus en el nostre codi, caldrà dedicar hores "extres" al projecte ja que aquesta fase serà la final i el marge de maniobra serà limitat.

2.4 Desviacions en la planificació

Les principals desviacions que ha sofert la planificació del projecte han estat sobretot en les fases de Preprocessament i de Creació i Avaluació del model . Seguidament s'expliquen les desviacions juntament amb el motiu que les ha provocat i les decisions que s'han pres en conseqüència:

- **Reducció de la inversió en temps en l'anàlisi descriptiva**: degut a que el client ja disposava d'informes amb l'anàlisi de les dades amb les quals es treballava, es va prendre la decisió d'escurçar el temps d'aquesta tasca. El motiu és que no s'ha considerat útil repetir la tasca d'anàlisi. La solució a aquest problema ha estat convocar una reunió amb el client on aquest ha exposat l'informe d'anàlisi de les dades i el desenvolupador l'ha utilitzat com a eina per a poder dur a terme les següents tasques.

- **Augment de temps dedicat a la tasca de preprocessament de les dades**: aquesta tasca ha sofert un augment de temps de 2 setmanes aproximadament. El motiu és que les dades tenien una complexitat més alta que la prevista i es tracta d'una tasca de gran importància per aconseguir un model predictiu de dades útil. S'ha decidit restar temps a la fase de Testing ja que es considera que la part

important del projecte és aconseguir un bon treball amb el model predictiu i no tant tenir una aplicació perfectament "polida". Al ser una prova de concepte, no és tant important la usabilitat i plena funcionalitat de l'aplicació sinó més focalitzar-se en la funcionalitat rellevant, que és la d'obtenir unes prediccions acurades.

- **Inversió de temps en la creació del servei que permet consultar el model de dades**: Inicialment estava previst utilitzar l'eina Azure Machine Learning Studio, no obstant, investigant i amb la col·laboració de desenvolupadors de client, es va optar per utilitzar l'eina Azure Machine Learning Workbench per a la creació i integració del model a l'aplicació. Aquesta eina permet una millor integració en tecnologies .NET i una personalització més alta dels servidors on s'allotgen els serveis. També ofereix altres avantatges que no han estat rellevants en el canvi de planificació.

Dins la tasca de Desenvolupament de la App , s'ha incorporat una nova subtasca per tal de poder allotjar el model a la plataforma Azure utilitzant l'eina descrita anteriorment. La planificació global de la tasca (en temps) no s'ha vist alterada, però s'ha cregut rellevant esmentar aquest canvi.

3 Gestió econòmica del projecte

En aquesta secció es descriuen els costos econòmics del projecte, tant els previstos com els inicials.

3.1 Estimació de costos del projecte

En aquesta secció es presentaran els costos estimats del projecte, ja siguin de Hardware, software i de mà d'obra o personal a banda dels costos indirectes(llum, gas aigua, etc.).

3.1.1 Costos de hardware

A la següent taula podem observar el hardware del qual requereix el projecte. Es compon principalment de l'ordinador per desenvolupar i els seus components.
Aquests costos són aplicables a totes les fases del projecte. Des de la primera fase fins a la darrera que es pot veure al diagrama de Gantt (Il·lustració 1: Diagrama de gantt del projecte) de la secció 2.

Producte	**Preu(€)**	**Unitats**	**Vida útil(anys)**	**Amortització(€)**
Dell Latitude E7470	1500	1	5	100
Acer K242HL	115	1	5	8
Logitech Wireless Mouse M185 Gris	13	1	5	1
Total:	1628		109	

Taula 3.1: Costos hardware

3.1.2 Costos de software.

A continuació es mostren els costos del software que s'utilitzarà durant el projecte.

Producte	**Preu(€)**	**Unitats**	**Vida útil(anys)**	**Amortització(€)**
Dell Latitude E7470	1500	1	5	100
Acer K242HL	115	1	5	8
Logitech Wireless Mouse M185 Gris	13	1	5	1
Total:	1628		109	

Taula 3.2: Costos software.

El cost del software Microsoft Azure va dedicat a les fases 3 i 5 del projecte. Per una banda a la fase 2 s'utilitzarà la tecnología Microsoft Azure Machine Learning mentre que a la fase 5 del diagrama de Gantt exposat a la secció 2 ja que s'utilitzaran els serveis al núvol creats en la paltaforma anterior. El cost associat al producte Visual Studio 2017 va relacionat sobretot amb la fase 5 del diagrama de Gantt esmentat anteriorment ja que será la plataforma utilitzada per a desenvolupar l'aplicació. El cost associat a Microsoft Office Professional 2016 s'utilitzarà durant totes les fases per documentar el projecte.

3.1.3 Costos de personal

El projecte consta principalment d'un desenvolupador. Aquest farà també de tester de l'aplicació. Per altra banda, el director de projectes s'encarregarà de dirigir el projecte i aconsellar-lo, fet que implica la destinació d'unes hores a aquesta tasca.

El director de projectes participarà en cadascuna de les tasques. La seva funció serà analitzar com es dur a terme cadascuna d'aquestes i si s'han de prendre determinades accions. S'entén que les hores dedicades es repartiran durant cada fase.

Rol	**Hores**	**€/hora**	**Salari(€)**
Desenvolupador	400	10	4000
Director de projectes	100	30	3000
Total:	500		7000

Taula 3.3: Costos de personal.

3.1.4 Costos no previstos

A la següent taula es mostren el màxim de costos extra previstos en el pitjor dels casos. Així doncs, podem tenir un marge que ens ajudi davant imprevistos.

Rol	**Hores**	**€/hora**	**Salari(€)**
Desenvoluapdor	20	10	200
Director de projectes	20	30	600
Total:	40		800

Taula 3.4: Previsió de costos no previstos

3.1.5 Costos Indirectes

A la següent taula es mostra una estimació dels costos indirectes:

Producte	**Preu**	**Unitats**	**Cost estimat(€)**
Electricitat	0.06€/kWh	2000 kWh	120
Internet	40€/mes	4 mesos	160
Aigua	22€/mes	4 mesos	88
Total:		368	

Taula 3.5: Costos indirectes previstos

3.1.6 Costos totals

A continuació es mostren els costos totals del projecte. Es té en compte un 5% de contingència per cobrir despeses derivades d'imprevistos:

Concepte	**Cost estimat(€)**
Hardware	109
Software	55
Personal	7000
No previstos	800
Indirectes	368
Subtotal: 8332	
Contingencia (5%): 416.6	
Total: 8748.6	

Taula 3.6: Costos totals

La següent taula mostra el desglossament del cost per tasca. El cost es calcula en funció de les hores dedicades a cada tasca. Aquests costos estan associats amb la planificació prevista a la secció 2:

Tasca	**% temps total**	**Cost estimat(€)**
Reconeixement de les dades	8	699
Anàlisi descriptiva	4	349
Preprocessament	16	1400
Creació i avaluació del model	16	1400
Aprenentatge de Xamarin	4	349
Desenvolupament de la App	48	4200
Introducció d'elements de Ludificació a la App	4	349
Testing	100	

Taula 3.7: Costos per tasca

3.2 Costos definitius del projecte

En aquesta secció es descriuen els costos definitius del projecte. Hi ha hagut una desviació clara en forma d'augment dels costos del projecte degut principalment a canvis en la planificació que han provocat un augment en el temps d'execució del projecte.
Cal tenir en compte que els costos tant de harware com de software no han sofert canvis ja que la seva vida útil i amortització estava prevista per un període de 5 anys, que, evidentment, és menor que el temps total de realització del projecte.

A la taula 3.8 es pot veure la comparativa entre els costos de personal inicials del projecte i els finals. Cal destacar que degut a l'augment de temps de realització del projecte, s'ha augmentat les hores del desenvolupador i en menor mesura, les del director de projectes ja que aquest no ha intervingut a les fases finals del projecte i ha anat cedint progressivament la responsabilitat al desenvolupador.

Rol	**Hores previstes**	**Hores finals**	**€/hora**	**Salari previst(€)**	**Salari final(€)**
Desenvolupador	400	600	10	4000	6000
Director de projectes	100	200	30	3000	6000
Total:	500	800	40	7000	12000

Taula 3.8: Costos de personal definitius vs costos de personal previstos

Per conseqüència, els costos indirectes (llum aigua, internet, etc.) han augmentat de forma proporcional al temps que ha augmentat la durada del projecte. A la taula 3.9 es mostra la taula actualitzada dels costos indirectes definitius.

Producte	**Preu**	**Unitats**	**Cost estimat(€)**
Electricitat	0.06€/kWh	2000 kWh	120
Internet	40€/mes	6 mesos	240
Aigua	22€/mes	6 mesos	132
Total:		502	

Taula 3.9: Costos indirectes finals

Finalment a les taules 3.10 i 3.11 es mostren els costos totals i dividits per tasques respectivament.

Concepte	**Cost estimat(€)**
Hardware	109
Software	55
Personal	12000
No previstos	800
Indirectes	502
Subtotal: 13466	
Contingencia (5%): 673.3	
Total: 14139.3	

Taula 3.10: Costos totals definitius

Tasca	**% temps total**	**Cost estimat(€)**
Reconeixement de les dades	8	1131.4
Anàlisi descriptiva	4	565.57
Preprocessament	16	2262.28
Creació i avaluació del model	32	4524.57
Aprenentatge de WPF	4	1131.4
Desenvolupament de la App	32	4524.57
Introducció d'elements de Ludificació a la App	4	1131.4

Taula 3.11: Costos segmentats per tasca definitius

4 Sostenibilitat i impacte social

En aquesta secció s'analitzarà l'impacte social i mediambiental que pot tenir el projecte a mes de com de sostenible pot ser. A continuació es mostra la matriu de sostenibilitat del projecte:

	PPP	**Vida útil**	**Riscs**
Ambiental	Consum del disseny	Emprempta ecològica	Riscs ambientals
	7/10	15/20	-2/-20
Econòmic	Factura	Pla de viabilitat	Riscs econòmics
	7/10	15/20	-12/-20
Social	Impacte personal	Impacte social	Riscs socials
	8/10	16/20	-4/-20
Rang de sostenibilitat	22/30	46/60	-16/-60
		32/90	

Taula 4.1: Matriu de sostenibilitat del projecte

4.1 Rang ambiental

Aquest projecte no requereix de grans recursos. Aquests recursos consumeixen bàsicament l'electricitat requerida per l'ordinador i perifèrics. Malgrat que s'intenta que aquests recursos no s'excedeixin molt, el fet de realitzar-se en una empresa on hi ha disponibles perifèrics com per exemple, monitors, el consum pugui ser una mica més de l'imprescindible.

L'impacte ambiental del projecte s'ha intentat reduir sobretot en la part d'allotjar el model Machine Learning a Azure (núvol). S'ha intentat demanar els mínims recursos possibles a la plataforma per tal de minimitzar els recursos hardware utilitzats.

Si es compta l'ús del portàtil com a ús principal i comptant que aquest està unes 8h/dia engegat. L'electricitat consumida al projecte és de és de 266.5 kw/h.

Si és realitzés de nou el projecte s'intentaria minimitzar el volum de dades utilitzat fer un anàlisi anterior per tal de minimitzar el cost computacional i d'aquesta forma reduir el consum energètic de l'equip. També es podria migrar la computació al núvol ja que, normalment, els recursos estan més optimitzats i per tant, es reduiria considerablement

el consum.

Malgrat existeixen solucions que es podrien adaptar al problema, el client vol una eina personalitzada i que treballi directament amb les seves dades. Per tant, malgrat que s'ha plantejat al client adaptar models de dades que treballin amb dades mediques, aquest ha decidit que no vol invertir recursos en l'adaptació.

Ambientalment aquest projecte no millorarà cap solució de les existents ja que aquesta no evita l'ús d'aparells electrònics. Malgrat això, tampoc implica haver de consumir nous aparells ja que l'aplicació es pot utilitzar en els ja existents.

Si el volum de dades augmentés significativament, s'haurien de consumir més recursos tant d'emmagatzematge com de computació. Aquest fet implicaria que la petjada ecològica augmentés significativament i és un risc que s'ha de tenir en compte. Una possible millora seria centrar molt més els esforços en la reducció de dades i centrar-se en emmagatzemar únicament la informació estrictament rellevant pel projecte.

4.2 Rang econòmic

Els costos del projecte s'han detallat de la manera més precisa possible a seccions anteriors. Aquest projecte requereix de consum elèctric i de connexió a internet ja que la interacció amb el client és primordial.
Els recursos personals utilitzats són els justos i imprescindibles. Només consta d'un desenvolupador i un director de projectes que s'intenta que intervingui el mínim d'hores possible.
El problema que es vol abordar és una solució que el client sol·licita. No hi ha una aplicació que faci exactament el mateix que la que es demana. Es podria intentar consultar un model existent per no haver de crear-lo, però la particularitat de les dades fa d'això una tasca difícil i que no s'estima que s'estalviïn recursos duent-la a terme.

El cost total del projecte és pot consultar a la taula 3.10. Degut a que els costos de personal en formen part el desenvolupador i el director de projectes, aquests estan limitats als sous d'aquests i, per tant, no es creu possible reduir-los. Per que fa als costos de material, ja s'ha previst utilitzar els mínims i imprescindibles (un ordinador i un ratolí bàsicament). Es podria haver optat per un ordinador més econòmic.

No es pot avaluar el cost del projecte durant la seva vida útil ja que es tracta d'una prova de concepte que en cap cas (com a tal) passaria a producció. Per avaluar-ho s'hauria de plantejar un possible escenari on posar en producció (quants usuaris en farien ús, quins sistemes utilitzarien, etc) per tal de poder avaluar el cost del projecte durant la vida útil.

4.3 Rang social

L'impacte personal que pot aportar el projecte al desenvolupador és molt important ja que aquest projecte té per objectiu facilitar la vida a gent que pateix malalties. A més a més. El fet de treballar amb tecnologies relacionades amb la intel·ligència artificial pot aportar una motivació extra al desenvolupador.

Socialment aquest projecte pot aportar una millora en la qualitat de vida dels usuaris ja que aquests poden adquirir mes coneixement sobre les seves dolències. A mes a mes, també a nivell psicològic pot animar a aquestes persones a prendre accions per millorar la seva condició i implicar-se més en la millora de la seva qualitat de vida.

Aquesta aplicació no pretén substituir els prescriptors sinó que pretén ser una eina d'ajuda també a aquests. Els prescriptors no hauran de fer tants esforços en analitzar els dades dels pacients. Això també implica que la part més monòtona i tediosa de les seves tasques es veurà reduïda.
La principal barrera d'aquesta aplicació és que no es preveu que sigui apta per a persones amb discapacitats visuals. El projecte es durà a terme amb llenguatge Python que és de software lliure i la plataforma Azure propietat de Microsoft. La decisió d'utilitzar aquesta última és pel motiu que ofereix un seguit d'eines que no estan disponibles amb software lliure.

El projecte pot millorar la qualitat de vida dels pacients. Això podria implicar una petita millora en la desigualtat social degut a que si es té una millor qualitat de vida és possible dur a terme tasques que no es poden dur a terme relacionades amb la mobilitat majoritàriament.

Com ja s'ha dit anteriorment, aquest projecte és una necessitat real del client i per tant, dur-lo a terme no és reinventar la roda ja que no existeixen eines "genèriques"que combinin tècniques de Ludificació i explotació de dades i que s'adaptin al context d'aquest projecte.

5 Identificació de lleis i regulacions

A continuació s'exposen les lleis i regulacions que s'han tingut en compte a l'hora de realitzar aquest projecte. Aquest projecte tracta amb dades personals de pacients per tal d'extreure'n informació i explotar-la per tal d'oferir una eina d'utilitat tant per als prescriptors com per als pacients.
A nivell de l'estat espanyol, que és la localització on es desenvolupa el projecte i es treballa amb les dades, la llei que regula la protecció de dades personals és la Llei orgànica de protecció de dades de caràcter personal (LOPD) [2]. L'objectiu d'aquesta llei és protegir les dades de caràcter personal dels pacients garantint-ne així la llibertat. A continuació es descriuen els diferents nivells de privacitat que estableix la LOPD:

- **Nivell bàsic**:
  - Identificatiu.
  - Personal.
  - Circumstàncies socials.
  - Acadèmics i professionals.
  - Detall d'empleats.
  - Informació comercial.
- **Nivell mitjà**:
  - Infraccions administratives o penals.
  - Hisenda pública.
  - Serveis Financers.
  - Solvència patrimonial i crèdits.
  - Avaluació de la personalitat.
- **Nivell alt**:
  - Ideologia.
  - Creences.
  - Origen racial.

Les principals dades que conté la base de dades i que podrien entrar en conflicte amb la llei LOPD, són principalment dades identificatives i de localització associades al nivell bàsic esmentat anteriorment. No es disposa, a la base de dades que s'ha utilitzat per a la realització del projecte, de dades associades al nivell mitjà ni al nivell alt.

S'ha tingut en compte aquesta llei durant l'execució del projecte ja que les dades que ens han cedit estan totalment anonimitzades. S'ha suprimit tant els noms com les adreces i els telèfons de contacte i, en general, totes les dades de caràcter identificatiu i personal. Per tant, els integrants del projecte no poden manipular cap dada de tipus personal garant així el compliment de la llei esmentada anteriorment.
Aquest fet no comporta cap impediment a l'hora de realitzar el projecte ja que les dades de caràcter personal no són necessàries per cap fase del projecte.

6 Desenvolupament del projecte

6.1 Descripció del model de dades

A continuació s'explicarà la base de dades de la qual es nodreix el projecte en general, que és de tipus relacional. És important entendre el model de dades ja que tota la implementació depèn d'aquest model.
La base de dades proporcionada té per objectiu modelar les relacions que existeixen entre els pacients i els seus tractaments, així com la relació entre pacients i personal mèdic. També permet emmagatzemar les mesures clíniques preses als pacients i així obtenir un històric d'aquestes. A més es registren les visites programades i realitzades pels pacients. Cal destacar que la base de dades proporcionades conté una gran quantitat de taules i relacions ja que aquesta està pensada per altres tipus d'aplicacions, com ara aplicacions de gestió de pacients, visualització de dades, etc. El primer objectiu és fer una selecció de les taules i relacions rellevants per al projecte i descartar totes aquelles.

6.1.1 Selecció dels elements rellevants del model

Tal com s'ha introduït anteriorment, cal fer una primera selecció dels elements(taules i relacions principalment) de la base de dades proporcionada. L'objectiu és evitar treballar amb dades irrellevants i reduir el volum de dades.
Tot seguit es descriuen les taules seleccionades. Es detallaran els camps rellevants així com la relació que tenen amb la resta de taules:

- **Patients**: aquesta taula identifica la informació bàsica d'un pacient (identificador, data de naixement, lloc de residència, etc.). Cada pacient pot no tenir assignats de 0 a infinits tractaments (taula Treatments).

- **Treatments**: descriu la informació principal d'un tractament mèdic. Aquest ha d'anar associat obligatòriament amb un pacient

- **TreatmentsHistory**: cada fila de la taula identifica un event (canvi, revisió, visita, etc.) produït en un tractament. Conté camps que identifiquen si s'ha produït algun canvi d'equipament, a quin grup pertany el tractament, el prescriptor que ha gestionat l'event.

- **EquipmentModels**: té com a finalitat emmagatzemar la informació de cadascun dels aparells utilitzats en un tractament. Van associats a la taula treatmentsHistory amb una relació (* - *).

- **TreatmentClinicalIndicatorsHistory**: serveix per modelar un identificador de tipus clínic. Els camps rellevants són la data de creació, el valor de l'indicador. Tot indicador clínic és alhora d'un tipus (taula ClinicalIndicatorTypes).

- **ClinicalIndicatorTypes**: Descriu un tipus d'identificador clínic. A la taula 6.1 es mostren uns quants exemples dels diferents tipus d'indicadors.

CIN_Id	CIN_Name	Description	CIN_UnitOfMeasure	CIN_MinInputValue	CIN_MaxInputValue
5	Use /WU	Hores utilitzades de la màscara de tipus WU.	h/j	0.00	3.00
6	Use (J>3h) WUC1	Funció WUC1 de la màscara utilitzada més de 3 hores al dia.	j(>3h)	0.00	7.00
7	Use (Ø) WUC2	Hores d'ús de la màscara de tipus WUC2.	h	0.00	24.00
8	Leaks (Ø) /WLR	Fuites de la màscara	l/min	0.00	99999999.00
9	IAH (Ø) /WIAH	Índex IAH.	e/h	0.00	40.00
10	Leaks (Ø) WLP	Fuites de tipus WLP.	%	0.00	99999999.00
11	Leaks (Ø) /WLS	Fuites de tipus WLS.	%	0.00	24.00
13	Utilisation (J>3h) /PUC1	Funció PUC1de la màscara utilitzada més de 3 hores al dia.	j(>3h)	0.00	7.00
15	Use (Ø) /PUA	Hores utilitzades de l'element de suport a la màscara.	h	0.00	24.00
18	Fuites (Ø) /PLR	Fuites de tipus PLR.	l/min	0.00	99999999.00

Taula 6.1: Tipus d'indicadors clínics.

- **TreatmentInterventions**: descriu un esdeveniment de tipus intervenció clínica. Els camps més rellevants són la data de creació, l'executor de la intervenció i el identicador del mode. Tota intervenció té un tipus (taula InterventionModes).

- **InterventionModes**: identifiquen el tipus d'intervenció clínica.

- **TreatmentProgressIndicatorsHistory**: identifica els indicadors de progres. Aquests estan encarats a emmagatzemar la informació de les medicions en les visites periòdiques. Els camps rellevants són: data de creació, valor, tipus (taula ProgessIndicatorTypes).

- **ProgressIndicatorTypes**: serveix per modelar els diferents tipus d'identificador de progrés. Descriuen els valors màxim i mínim que pot tenir el mesurament d'aquest tipus. A la taula 6.2 es mostren uns quants exemples dels indicadors de progrés.

PIT_Id	Name	PIT_DefaultUnitOfMeasure	PIT_MinValue	PIT_MaxValue
2	P90/95	cm H2O	0.00	50.00
4	Residual IAH	Evmts/h	0.00	150.00
6	Initial IAH	Evmts/h	0.00	250.00
15	SpO2 without O2	%	50.00	100.00
16	SpO2 with O2	%	50.00	100.00
23	Weight	Kg	0.50	250.00
33	Vol Hydrat	ml	0.00	9999.00
40	Body temperature	C	35.00	45.00

Taula 6.2: Tipus d'indicadors de progrés.

- **TreatmentRiskFactorHistory**: fan referencia als indicadors de risc que sorgeixen al llarg d'un tractament. Emmagatzemen el valor de risc detectat en una determinada data i poden ser de diferents tipus (taula RiskFactorTypes). Aquest

poden ser de perill alt, mitjà o baix i el tipus de tractament indica els rangs on aquest és d'un determinat perill. Cada element d'aquesta taula té un tipus assignat.

- **RiskFactorTypes**: idèntica un tipus d'identificador de risc. Aquesta taula indica el rang de valors perquè una fila de la taula TreatmentRiskFactorHistory sigui de valor baix mitjà o alt. A la Il·lustració 1 es pot observar el diagrama en format UML del model que representa de forma gràfica l'explicació anterior. Cal remarcar que a cada taula només els mostren els camps que fan referència a claus, ja siguin foranes o bé primàries.

A la Il·lustració 6.1 es pot observar el diagrama en format UML del model que representa de forma gràfica l'explicació anterior. Cal remarcar que a cada taula només els mostren els camps que fan referència a claus, ja siguin foranes o bé primàries.

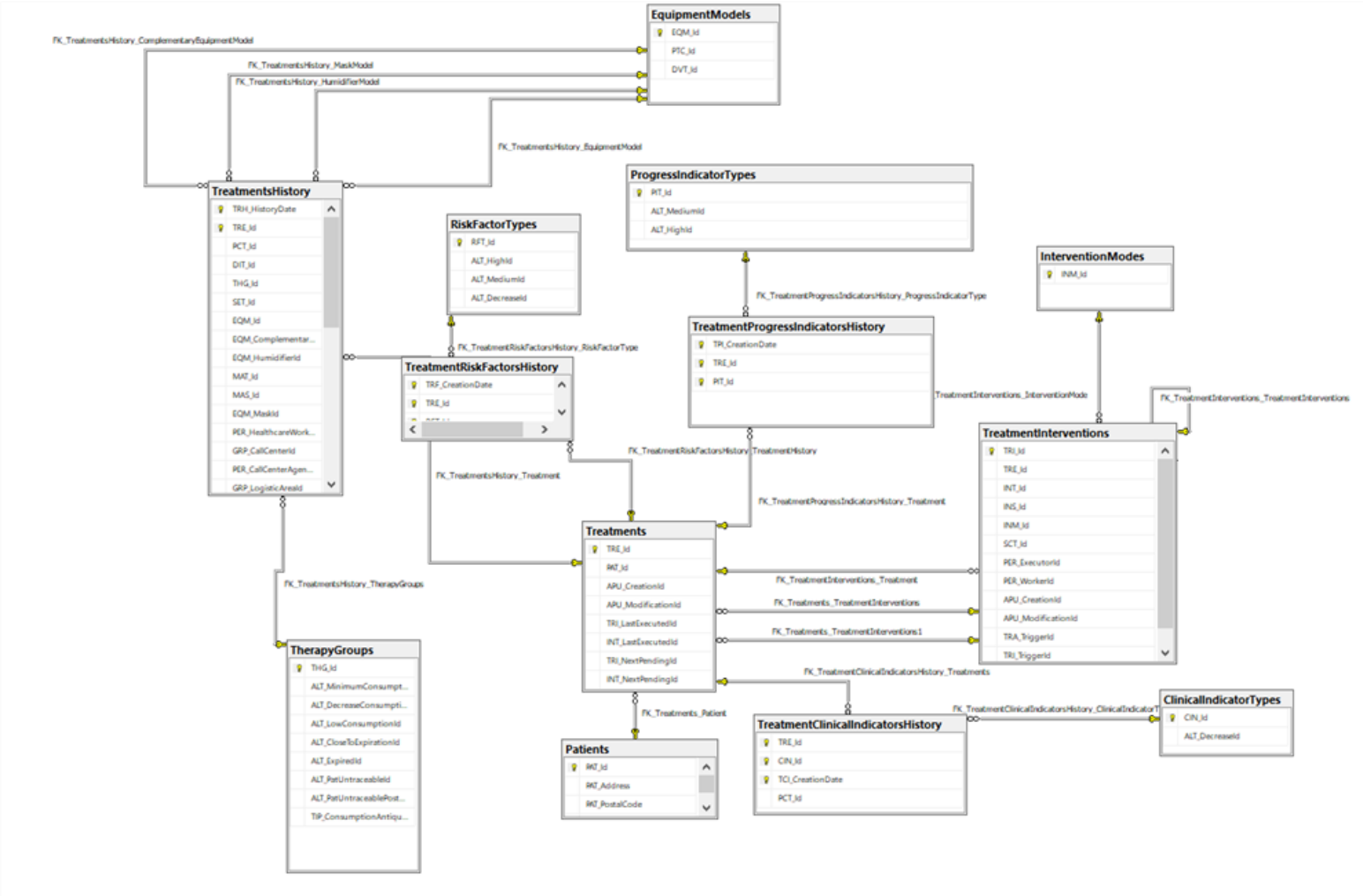

Figura 6.1: Diagrama UML del model

6.2 Transformació de les dades

L'objectiu d'aquesta fase és adaptar les dades al problema que volem afrontar.
Els processos que apliquen tècniques de Machine Learning requereixen d'un format de dades específic ja que treballar directament amb un model relacional sería una tasca molt costosa i perjudicaria tant el rendiment com el desenvolupament i avaluació del model.
En el problema que es vol abordar requereix d'aplicar tècniques clàssiques de amb Machine Learning , aquestes generalment requereixen treballar amb dades en formats estàndard com són fulls de càlcul (.csv) o formats equivalents o similars.
Tenint en compte l'esmentat anteriorment el primer objectiu en aquesta fase és transformar les dades per tal de passar d'un format relacional (Base de dades SQL) a un format que vàlid per a abordar el problema.
Un altre característica rellevant dels problemes d'anàlisi i explotació de dades en general és que requereixen d'una gran càrrega computacional degut a que treballen amb un volum elevat de dades. Aquest fet implica la necessitat de fer una selecció de les dades que siguin rellevants i eliminar totes aquelles duplicades o redundants. Per exemple, cal eliminar les variables que no tenen incidència en la variable predictiva i també eliminar totes aquelles columnes duplicades o que no aporten valor al problema.

6.2.1 Transformació i selecció de les taules

Com ja s'ha introduït anteriorment, el primer pas que cal dur a terme és el de transformar el model de dades relacional a un format apte per aplicar les tècniques de Machine Learning.
La base de dades proporcionada està desenvolupada amb la tecnologia Microsoft SQL-Server [8] que utilitza un format similar a l'estàndard per bases de dades (SQL).

El primer pas que s'ha realitzat és el de seleccionar les taules rellevants del model. Per abordar el problema, cal tenir en compte com s'estructura el model de dades amb el qual s'ha treballat.

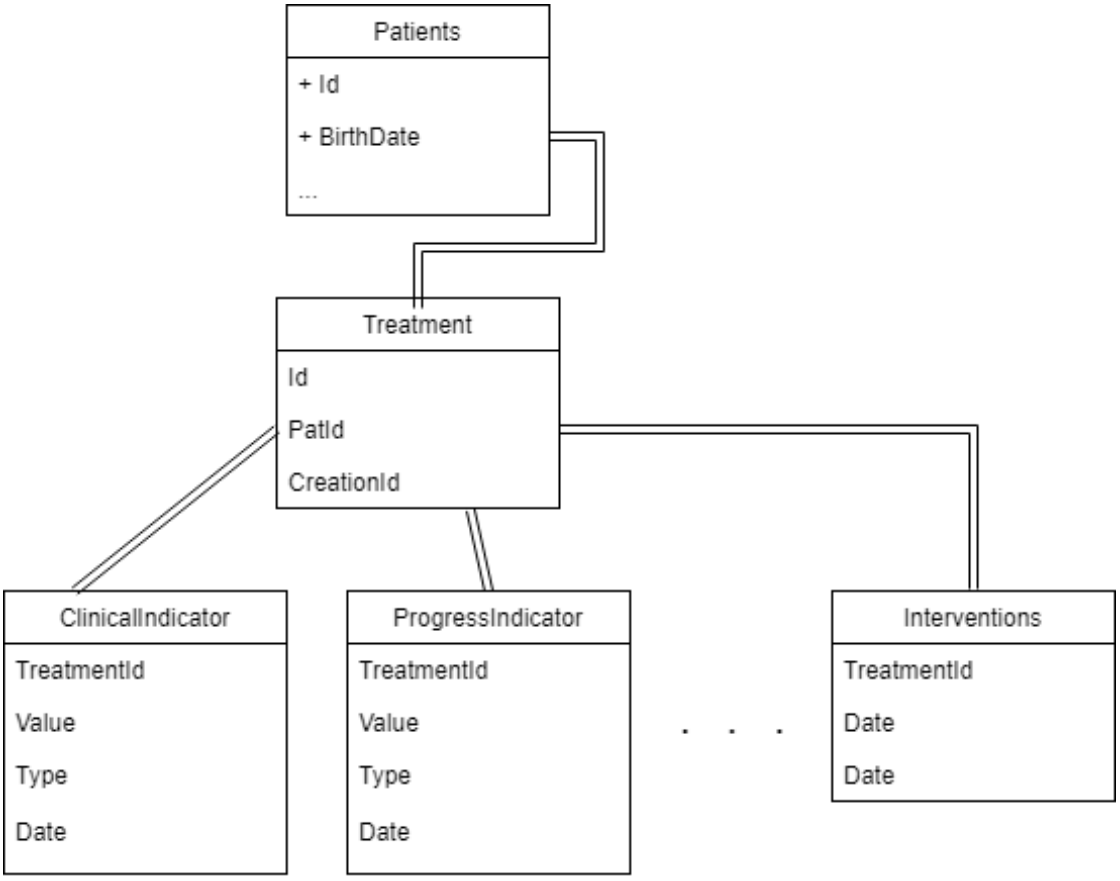

Figura 6.2: Esquema general del model de dades (taules seleccionades)

La il·lustració 6.2 mostra un esquema general de les taules seleccionades. El model conté pacients i tractaments on cada tractament està associat a un pacient. A més, cada tractament té associades diferents taules que fan referència a indicadors clínics, indicadors de progrés, intervencions etc. que ocorren durant l'execució del tractament. Aquests elements són rellevants ja que ens permeten obtenir informació sobre l'evolució dels tractaments. La resta de taules no ho són ja que fan referència a aspectes concrets d'altres aplicacions que utilitzen la mateixa base de dades i no aporten informació que es consideri rellevant pel problema que s'ha abordat.

6.2.1.1 Preprocessament individual de les taules

Abans d'agrupar les taules conjuntament per tal de tenir un model usable, cal primer treballar amb les taules seleccionades individualment per tal d'eliminar-ne la informació no-rellevant.

Degut a que el volum de dades és molt elevat, s'ha pres la decisió d'agrupar les taules que fan referencia a indicadors dels tractaments per setmana [1] i no per data exacta tal

[1]Per calcular la setmana s'ha pres com a referencia la primera data de les quals es disposa de da-

com estan modelades inicialment. Per a cada camp que indica un valor, s'ha pres com a valor la mitjana durant la setmana. Les taules a les quals se'ls ha aplicat aquesta transformació són: TreatmentClinicalIndicatorsHistory, TreatmentProgresIndicatorsHistory, TreatmentsConsumptionsHistory, TreatmentRiskFactorHistory, i Treatmentinterventions.

El procés de transformació consisteix a llegir les taules des de la base de dades (SQLServer) i generar un fitxer de càlcul (en format .csv) apte per a aplicar tècniques de *Machine Learning*.
Aquest procés es realitzarà mitjant scripts en el llenguatge Python [9] i utilitzant principalment les següents llibreries:

- Pandas [10]: Llibreria d'anàlisi de dades en python. Ens permet llegir dades en qualssevol format(full de càlcul, SQL) i convertir modelar les dades llegides.

- Pyodbc [11]: és un mòdul Python de codi obert que facilita l'accés a bases de dades ODBC. S'utilitza per consultar les dades des de la base de dades SQLServer.

La il·lustració 6.12 mostra un esquema general del procés.

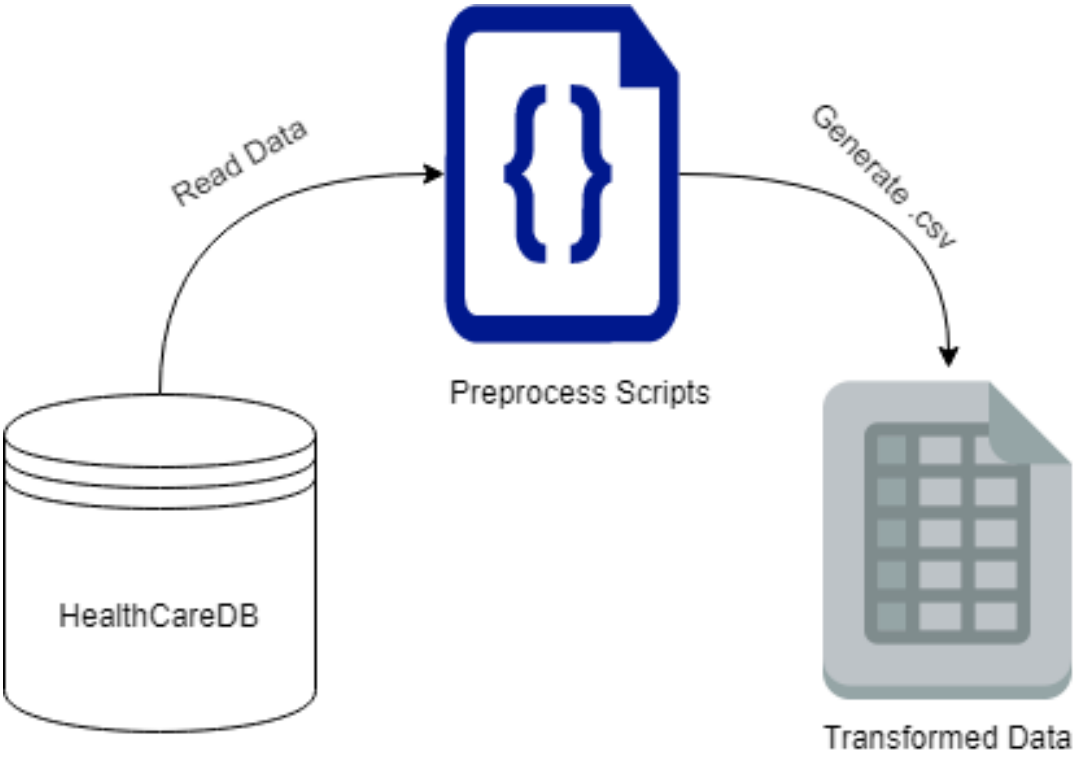

Figura 6.3: Esquema de la fase de preprocés

des(17/6/2009)

En els següents apartats s'expliquen les transformacions realitzades a cadascuna de les taules seleccionades.

6.2.1.1.1 Patients

Aquesta taula conté i modela la informació bàsica dels pacients.
Els camps rellevants són els següents:

- PAT_Id: identificador del pacient.
- PAT_BirthDate: data de naixement del pacient.
- PAT_Gender: gènere (Masculí o femení).

Per aquesta taula únicament s'han eliminat les columnes no-rellevants i s'ha calculat l'edat del pacient(partint com a data de referencia el dia: 27/11/2017).

6.2.1.1.2 Treatments

Aquesta taula modela i conté la informació bàsica d'un tractament. Cada tractament esta associat a un pacient.
Els camps rellevants són els següents:

- TRE_Id: identificador del tractament.
- PAT_Id: pacient associat al tractament.

Per aquesta taula únicament s'han eliminat les columnes no-rellevants.

6.2.1.1.3 TreatmentClinicalIndicatorsHistory

Aquesta taula fa referència als indicadors clínics associats a cada tractament. Tot indicador té un tipus (taula ClinicalIndicatorTypes). A més d'agrupar les files d'aquesta taula per setmana, també les agruparem per tipus ja que ens interessa distingir els diferents tipus d'indicadors clínics associats a un pacient [2].
Els camps rellevants d'aquesta taula són:

- TCI_NumericComputedValue: aquest camp indica el valor numèric de l'indicador, independentment de si el tipus és numèric o no, aquest valor sempre ho serà. Seleccionem aquest camp i no TCI_Value ja que aquest útil no té perqué contindre un valor numèric i el primer ens estalvia una futura transofrmació.
- TCI_CreationDate: la data de creació de l'element és rellevant ja que ens permet després obtenir la setmana de creació de l'indicador.

[2] L'objectiu de la fase de crear el model, és fer un model predictiu que sigui capaç de predir el valor dels indicadors clínics de tipus 6

- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.
- CIN_Id: aquest camp ens identifica el tipus d'indicador clínic.

Un cop esmentats els camps que s'han considerat rellevants en aquesta taula, a continuació es defineixen els passos de preprocés realitzats per aquesta taula:

- Eliminar els camps no rellevants de les taules (tots els que no s'han esmentat anteriorment).
- Calcular la setmana de creació per cada element. El següent codi mostra com s'ha calculat:

```
#datefrom = 17/6/2009
    start_date_monday = (dateFrom - DT.timedelta(days=dateFrom.weekday()))
    num_of_weeks = math.ceil((dateTo - start_date_monday).days / 7.0)
    return num_of_weeks
```

- Agrupar per [TRE_Id,Setmana] i pivotar la taula per tal de tenir els valors per setmana de cadascun dels tipus. Per exemple:

TRE_Id	**WeekCreated**	**CIN_Value_1**	**CIN_Value_2**	**...**	**CIN_Value20**
2	2	1	2	**...**	4
2	3	1	3	...	3

Taula 6.3: Exemple de la taula d'indicadors després d'aplicar el procés de "pivotatge".

6.2.1.1.4 TreatmentProgresIndicatorsHistory

Aquesta taula fa referència als indicadors de progrés associats a un determinat tractament. Tot indicador de progrés té un tipus (taula ProgresIndicatorTypes).
Els camps rellevants d'aquesta taula són:

- TPI_NumericComputedValue: aquest camp indica el valor numèric de l'indicador, independentment de si el tipus és numèric o no, aquest valor sempre ho serà.
- TPI_CreationDate: la data de creació de l'element és rellevant ja que ens permet després obtenir la setmana de creació de l'indicador.
- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.
- PIT_Id: aquest camp ens identifica el tipus d'indicador clínic. [3]

Els passos de preprocés són els mateixos que per a la taula anterior (secció 6.2.1.1.3) però amb els camps corresponents a aquesta taula.

[3] Per a aquest problemas'han seleccionat només els indicadors de progrés de tipus numèric.

6.2.1.1.5 TreatmentRiskFactorHistory

Aquesta taula fa referència als indicadors de risk associats a un determinat tractament. Tot indicador de risk té un tipus (taula RiskFactorTypes).

Els camps rellevants d'aquesta taula són:

- TRF_Value: aquest camp indica el valor numèric de l'indicador, independentment de si el tipus és numèric o no, aquest valor sempre ho serà.
- TRF_CreationDate: la data de creació de l'element és rellevant ja que ens permet després obtenir la setmana de creació de l'indicador.
- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.
- RFT_Id: aquest camp ens identifica el tipus d'indicador de progrés. [4]

6.2.1.1.6 TreatmentsHistory

Aquesta taula serveix per modelar l'històric general de cada tractament. Per exemple ens indica si s'han realitzat canvis en l'equipament medic del pacient, si hi hagut canvi de prescriptor, etc.

Els camps rellevants d'aquesta taula són:

- TRH_CreationDate: la data de creació de l'element és rellevant ja que ens permet després obtenir la setmana de creació de l'indicador.
- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.
- TRH_EquipmentId: identificador de l'equipament que porta el pacient. S'utilitzarà per comptar quants canvis d'equipament ha sofert el pacient durant cada setmana (1 si no hi ha hagut canvis, 2 si n'hi hagut un, etc.).
- TRH_Mask: identificador de la màscara d'oxigen que porta el pacient. S'utilitzarà per comptar quants canvis de mascara ha sofert el pacient durant cada setmana (1 si no hi ha hagut canvis, 2 si n'hi hagut un, etc.).
- PER_PrescriberId: identificador del prescriptor del tractament. S'utilitzarà per comptar quants canvis de prescriptor ha sofert el tractament durant cada setmana (1 si no hi ha hagut canvis, 2 si n'hi hagut un, etc.).

Els passos de preprocés d'aquesta taula són els següents:

1. Eliminar els camps que no són rellevants.
2. Calcular la setmana per cada fila de la taula.

[4] Per a aquest problema s'ha decidit seleccionar només els indicadors de risc de tipus numèric.

3. Agrupar per [TRE_Id, Setmana].
4. Per cada agrupació, calcular el nombre de diferents valors dels camps: TRH_EquipmentId, TRH_Mask i PER_PrescriberId. Eliminar finalment els duplicats.
5. Renombrar les columnes anteriors com: [TRH_EquipmentChanges, TRH_MaskChanges, TRH_Prescribers]

A la següent taula es mostra un exemple de com queda la taula un cop aplicades les transformacions:

TRE_Id	**WeekCreated**	**TRH_EquipmentChanges**	**TRH_MaskChanges**	**TRH_Prescribers**
2	2	1	2	**1**
2	3	1	3	2

Taula 6.4: Taula TreatmentsHistory després d'aplicar-ne les transformacions

6.2.1.1.7 TreatmentInterventions

Aquesta taula té per objectiu modelar les intervencions produïdes durant l'execució del tractament.
Els atributs que s'han considerat rellevants d'aquesta taula són:

- TRI_PlannedDate: data d'execució de la intervenció.
- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.

L'objectiu és identificar el nombre d'intervencions setmanals associades a cada tractament.

Els passos de preprocés a seguir són:

- Eliminar els atributs no rellevants.
- Calcular la setmana per cada fila de la taula.
- Agrupar per [TRE_Id, Setmana].
- Afegir una nova columna que sigui el resultat de comptar el nombre d'intervencions setmanals.

A la següent taula es mostra un exemple de com queda la taula un cop aplicades les transformacions:

TRE_Id	**WeekCreated**	**TRH_EquipmentChanges**	**TRH_MaskChanges**	**TRH_Prescribers**
2	2	1	2	**1**
2	3	1	3	2

Taula 6.5: Taula TreatmentInterventions després d'aplicar-ne les transformacions.

6.2.1.1.8 TreatmentConsumptionHistory

Aquesta taula té per objectiu modelar les mesures de consum preses al pacient (en un tractament en concret).

Els camps rellevants d'aquesta taula són:

- TRH_ReadingDate: la data de mesura de l'element és rellevant ja que ens permet després obtenir la setmana de creació de l'indicador.
- TRE_Id: identifica el tractament al qual l'indicador clínic està associat.
- TCH_ReadingValue: valor consumit (litres) des de la ultima data de mesura.
- TCH_PreviousReadingValue: última mesura presa abans de l'actual.
- TCH_DaysBetweenReadings: dies que han passat entre l'última mesura i l'actual.
- TCH_Ratio: camp calculat amb la següent fòrmula:
  (TCH_ReadingValue - TCH_PreviousReadingValue) / TCH_DaysBetweenReadings.
  Aquest camp permet normalitzar el valor de lectura.

Un cop esmentats els camps rellevants, els passos de preprocés per aquesta taula són:

- Eliminar els atributs no-rellevants.
- Calcular la setmana a partir del camp TCH_ReadingValue.
- Agrupar per [TRE_Id, Setmana] i calcular la mitjana del camp TCH_Ratio per cadascun del grups.

A la següent taula es mostra un exemple de com queda la taula un cop aplicades les transformacions:

TRE_Id	**WeekCreated**	**TCH_Ratio**
2	3	2.23
2	4	3.34

Taula 6.6: Taula TreatmentConsumptionsHistory després d'aplicar-ne les transformacions

6.2.1.2 Agrupament de les taules

Un cop fet el preprocés de les taules rellevants de forma individual, el següent pas consisteix a agrupar les dades per tal de crear un únic fitxer (en format .csv). Per a fer-ho, s'ha decidit fer les següents agrupacions de les dades:

1. Agrupar les taules Patients i treatments a partir del camp comú TRE_Id. A l'estructura de dades sorgida l'anomenarem PT.

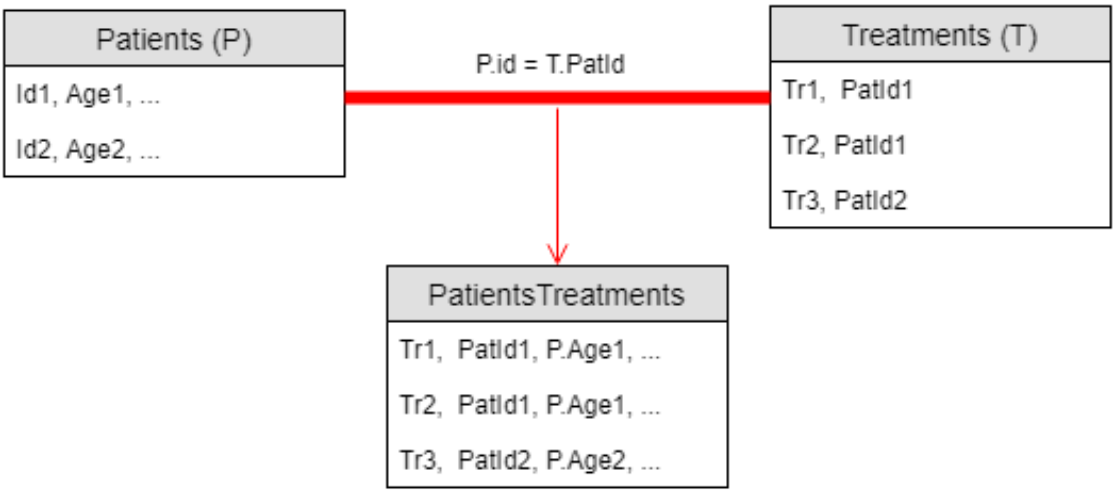

Figura 6.4: Exemple d'agrupació entre les taules Patients i Treatments

2. Agrupar les taules que fan referència a indicadors de tractaments(TreatmentsHistory, TreatmentRiskFactorHistory, TreatmentClinicalIndicators, TreatmentInterventions, TreatmentConsumptionsHistory, TreatmentProgressIndicators) pels camps: [TRE_Id,WeekCreated] que contenen totes les taules esmentades. A l'estructura de dades sorgida es denominarà JI.

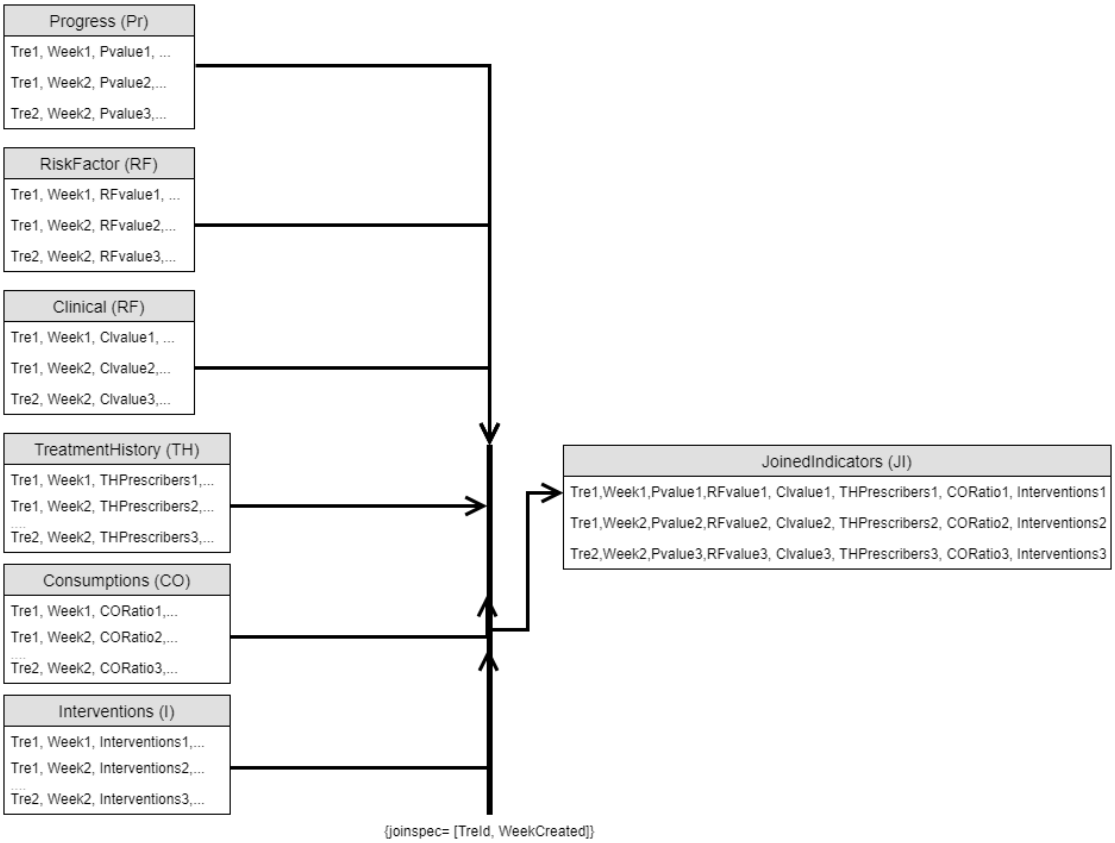

Figura 6.5: Exemple d'agrupació entre les taules d'indicadors de tractaments.

3. L'últim pas és agrupar les estructures de dades generades en els dos passos anterior (PT i JI) pel camp [TRE_Id]. A aquesta taula generada l'anomenarem DataSet.

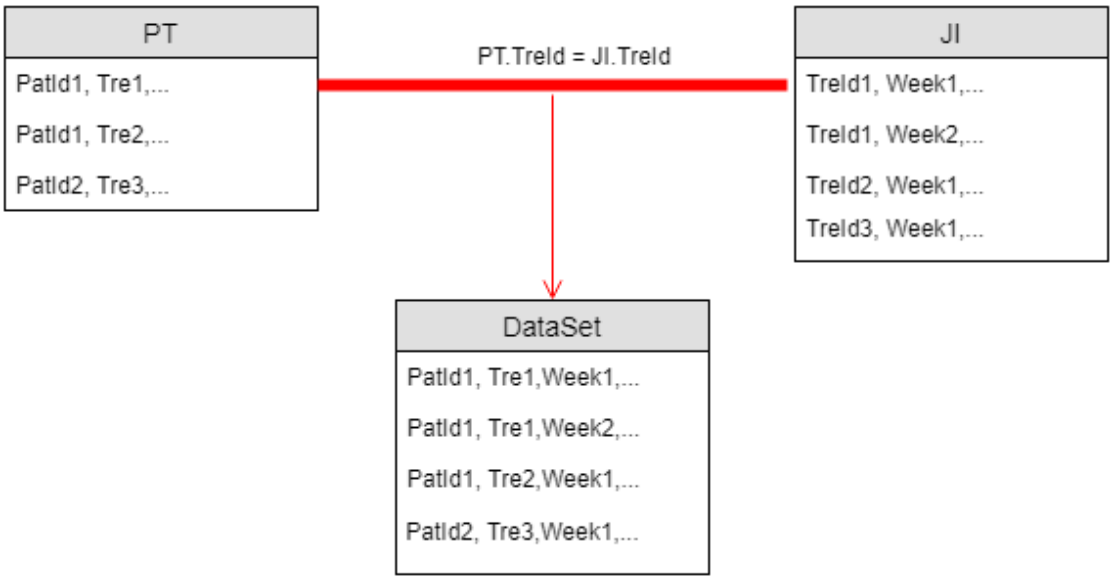

Figura 6.6: Exemple de la taula generada un cop aplicats el tercer pas.

Un cop aplicats aquests passos s'ha generat un fitxer en format full de càlcul (.csv) que és resultat d'aplicar els passos esmentats anteriorment.

6.3 Anàlisi de dades

Cal remarcar que les dades encara no estan preparades per treballar amb mètodes de Machine Learning en aquest punt ja que abans cal assegurar-se de que les dades no s'hi troben valors nuls, outliers, etc. Abans d'analitzar els passos a seguir per tenir unes dades netejades, en la següent secció s'analitzaran les dades obtingudes al moment.

En seccions anteriors, hem vist que hem transformat les dades inicials provinents d'un model relacional a un fitxer de càlcul(.csv). A continuació es mostra un anàlisi de les dades que conté aquest fitxer:

- Files: 5413918 .
- Columnes: 89. Les columnes són:
  - Id del pacient.
  - Id del tractament.
  - Sexe del pacient.
  - Setmana.
  - Mitjana setmanal per cadascun dels tipus de clinicalIndicators
  - Mitjana setmanal per cadascun dels tipus de progresIndicators .

- Mitjana setmanal per cadascun dels tipus de riskFactors .
- Mitjana setmanal del ratio de consum.
- Nombre d'intervencions setmanals.
- Nombre de canvis de prescriptor setmanals.

A la següent figura es poden veure els noms de totes les columnes:

```
Index(['PAT_Id', 'TRE_Id', 'PAT_Age', 'PAT_Gender', 'WeekCreated',
       'InterventionsWeek', 'ClinicalValue_5', 'ClinicalValue_6',
       'ClinicalValue_7', 'ClinicalValue_8', 'ClinicalValue_9',
       'ClinicalValue_10', 'ClinicalValue_11', 'ClinicalValue_12',
       'ClinicalValue_13', 'ClinicalValue_14', 'ClinicalValue_15',
       'ClinicalValue_17', 'ClinicalValue_18', 'ClinicalValue_19',
       'ClinicalValue_20', 'ProgressValue_2', 'ProgressValue_4',
       'ProgressValue_6', 'ProgressValue_7', 'ProgressValue_11',
       'ProgressValue_12', 'ProgressValue_13', 'ProgressValue_15',
       'ProgressValue_16', 'ProgressValue_21', 'ProgressValue_22',
       'ProgressValue_23', 'ProgressValue_25', 'ProgressValue_26',
       'ProgressValue_27', 'ProgressValue_29', 'ProgressValue_30',
       'ProgressValue_33', 'ProgressValue_34', 'ProgressValue_35',
       'ProgressValue_36', 'ProgressValue_38', 'ProgressValue_40',
       'ProgressValue_41', 'ProgressValue_45', 'ProgressValue_46',
       'ProgressValue_47', 'ProgressValue_48', 'ProgressValue_53',
       'ProgressValue_54', 'ProgressValue_55', 'ProgressValue_56',
       'ProgressValue_57', 'ProgressValue_58', 'RiskValue_1', 'RiskValue_2',
       'RiskValue_3', 'RiskValue_4', 'RiskValue_5', 'RiskValue_6',
       'RiskValue_7', 'RiskValue_8', 'RiskValue_9', 'RiskValue_10',
       'RiskValue_11', 'RiskValue_12', 'RiskValue_13', 'RiskValue_14',
       'RiskValue_15', 'RiskValue_16', 'RiskValue_17', 'RiskValue_18',
       'RiskValue_19', 'RiskValue_20', 'RiskValue_21', 'RiskValue_22',
       'RiskValue_23', 'RiskValue_24', 'RiskValue_25', 'RiskValue_26',
       'RiskValue_28', 'RiskValue_29', 'RiskValue_30', 'RiskValue_33',
       'RiskValue_34', 'TRH_EquipmentChanges', 'TRH_Prescribers',
       'Consumption', 'Ratio'],
```

Figura 6.7: Noms de les columnes del conjunt de dades abans d'aplicar els passos de preprocessat.

És important analitzar la qualitat de les dades, és a dir, veure si existeixen valors anómals, si hi ha outliers a les dades, etc. S'ha de tenir en compte que apareixen força valors nuls ja que al fer les agrupacions a les dades, hi ha setmanes on no existeixen valors per a alguna de les columnes. Ja que disposem d'un gran volum de dades i aplicar tècniques de substitució de valors podria comportar un gran cost computacional, s'ha decidit eliminar les columnes en que existeix algun valor nul.

Una altra decisió que s'ha pres és la d'eliminar les files on la setmana de creació sigui major a la 206 (corresponen a la setmana del 17//2013). Aquest fet implica que les dades de prova aniran de la setmana següent a aquesta fins a la 417 (corresponent al 16/3/2017). Aquesta decisió és deguda a que per al desenvolupament de l'aplicació, es necessiten dades de prova amb les quals el model predictiu no ha de treballar.

El nombre de files un cop aplicat aquesta passos és de 1627903, aproximadament el 30% de l'original. Aquest fet indica que les dades tenen una elevada dispersió en el temps. Això és perquè no cada setmana es prenen les mesures per a tots els indicadors.

A continuació es mostrarà l'anàlisi realitzat de les dades un cop eliminats els valors nuls.

	count	mean	std	min	25%	50%	75%	max
PAT_Id	2717621.0	38295.86	28342.56	1.0	14161.0	35010.0	56510.0	125342.0
TRE_Id	2717614.0	84103.27	50830.43	1.0	48778.0	85025.0	106788.0	207861.0
PAT_Age	2717621.0	65.45	15.43	0.0	58.0	67.0	75.0	365.0
WeekCreated	2688254.0	279.38	67.72	2.0	252.0	294.0	330.0	366.0
InterventionsWeek	594377.0	1.16	0.51	0.0	1.0	1.0	1.0	24.0
ClinicalValue_5	1630018.0	1.37	0.9	0.0	0.0	2.0	2.0	4.0
ClinicalValue_6	1630018.0	4.78	2.83	0.0	2.0	6.0	7.0	14.0
ClinicalValue_7	1630018.0	4.55	3.02	0.0	2.0	5.0	7.0	24.0
ClinicalValue_8	1630018.0	2.81	7.8	0.0	0.0	0.0	2.4	198.0
ClinicalValue_9	1630018.0	2.09	28.76	0.0	0.0	1.0	3.0	25719.0
ClinicalValue_10	1630018.0	0.17	1.87	0.0	0.0	0.0	0.0	99.0
ClinicalValue_11	1630018.0	0.58	4.03	0.0	0.0	0.0	0.0	100.0
ClinicalValue_13	1630018.0	4.77	9.91	0.0	0.0	0.0	0.0	28.0
ClinicalValue_15	1630018.0	11.31	5305.42	0.0	0.0	0.0	0.0	2834498.0
ClinicalValue_17	1630018.0	0.53	7.39	0.0	0.0	0.0	0.0	6433.0
ProgressValue_2	244154.0	4.29	5.87	-8.9	0.0	0.0	9.8	1038.0
ProgressValue_4	244154.0	19.96	31.84	0.0	0.0	26.12	32.41	4429.0
ProgressValue_6	244154.0	3.64	9.61	0.0	0.0	0.0	0.0	312.0
ProgressValue_7	244154.0	-2930.8	1448694.09	-715827882.33	0.0	0.0	1.2	168.67
ProgressValue_11	244154.0	0.23	1.8	0.0	0.0	0.0	0.0	115.6
ProgressValue_12	244154.0	5.63	41.66	0.0	0.0	0.0	0.0	14158.44
ProgressValue_13	244154.0	1.21	6.38	0.0	0.0	0.0	0.0	554.25
ProgressValue_15	244154.0	26.75	46.6	0.0	0.0	0.0	84.0	1851.0
ProgressValue_16	244154.0	19.88	39.45	0.0	0.0	0.0	45.0	1995.0
ProgressValue_21	244154.0	0.14	3.32	0.0	0.0	0.0	0.0	162.0
ProgressValue_22	244154.0	0.1	3.22	0.0	0.0	0.0	0.0	346.0
ProgressValue_23	244154.0	0.52	6.2	0.0	0.0	0.0	0.0	202.0
ProgressValue_25	244154.0	0.19	2.81	0.0	0.0	0.0	0.0	578.51
ProgressValue_26	244154.0	3.67	100.37	0.0	0.0	0.0	0.0	6300.0

Table 6.7 Continua des de la pàgina anterior

ProgressValue_27	244154.0	0.55	41.42	0.0	0.0	0.0	0.0	4795.0
ProgressValue_29	244154.0	0.11	3.72	0.0	0.0	0.0	0.0	397.33
ProgressValue_30	244154.0	0.0	0.28	0.0	0.0	0.0	0.0	39.75
ProgressValue_33	244154.0	6.64	121.55	0.0	0.0	0.0	0.0	20000.0
ProgressValue_34	244154.0	0.0	0.51	0.0	0.0	0.0	0.0	84.0
ProgressValue_35	244154.0	0.0	0.01	0.0	0.0	0.0	0.0	2.74
ProgressValue_36	244154.0	0.0	0.1	0.0	0.0	0.0	0.0	35.1
ProgressValue_38	244154.0	0.0	0.03	0.0	0.0	0.0	0.0	10.4
ProgressValue_40	244154.0	0.01	0.32	0.0	0.0	0.0	0.0	70.0
ProgressValue_41	244154.0	0.0	0.05	0.0	0.0	0.0	0.0	7.0
ProgressValue_45	244154.0	0.06	2.09	0.0	0.0	0.0	0.0	624.0
ProgressValue_46	244154.0	0.09	2.24	0.0	0.0	0.0	0.0	148.6
ProgressValue_48	244154.0	0.79	5.09	0.0	0.0	0.0	0.0	98.0
ProgressValue_53	244154.0	0.33	8.85	0.0	0.0	0.0	0.0	930.0
ProgressValue_54	244154.0	0.1	2.89	0.0	0.0	0.0	0.0	350.0
ProgressValue_55	244154.0	0.0	0.35	0.0	0.0	0.0	0.0	96.0
ProgressValue_56	244154.0	0.09	3.31	0.0	0.0	0.0	0.0	400.0
ProgressValue_57	244154.0	0.09	3.46	0.0	0.0	0.0	0.0	480.0
ProgressValue_58	244154.0	0.0	0.09	0.0	0.0	0.0	0.0	10.0
RiskValue_1	184630.0	2.36	2.61	0.0	0.0	1.0	5.0	20.0
RiskValue_2	184630.0	418.17	196.31	0.0	375.0	500.0	525.0	3200.0
RiskValue_3	184630.0	0.08	0.63	0.0	0.0	0.0	0.0	20.0
RiskValue_4	184630.0	3.06	2.33	0.0	0.0	5.0	5.0	26.0
RiskValue_5	184630.0	3.12	20.07	-7.0	0.0	2.0	5.0	5899.0
RiskValue_6	184630.0	0.33	1.08	0.0	0.0	0.0	0.0	13.0
RiskValue_7	184630.0	0.1	0.41	0.0	0.0	0.0	0.0	12.0
RiskValue_8	184630.0	0.05	0.22	0.0	0.0	0.0	0.0	4.0
RiskValue_9	184630.0	0.88	0.89	0.0	0.0	1.0	1.0	8.0
RiskValue_10	184630.0	0.28	0.9	0.0	0.0	0.0	0.0	8.0

Table 6.7 Continua des de la pàgina anterior

RiskValue_11	184630.0	0.03	0.4	0.0	0.0	0.0	0.0	24.0
RiskValue_12	184630.0	0.0	0.03	0.0	0.0	0.0	0.0	4.0
RiskValue_13	184630.0	0.0	0.08	0.0	0.0	0.0	0.0	7.0
RiskValue_14	184630.0	0.0	0.1	0.0	0.0	0.0	0.0	8.0
RiskValue_15	184630.0	0.03	0.25	0.0	0.0	0.0	0.0	8.0
RiskValue_16	184630.0	0.0	0.16	0.0	0.0	0.0	0.0	50.0
RiskValue_17	184630.0	0.03	0.25	0.0	0.0	0.0	0.0	8.0
RiskValue_18	184630.0	0.05	1.26	0.0	0.0	0.0	0.0	500.0
RiskValue_19	184630.0	0.0	0.09	0.0	0.0	0.0	0.0	6.0
RiskValue_20	184630.0	0.06	1.27	0.0	0.0	0.0	0.0	500.0
RiskValue_21	184630.0	0.01	0.16	0.0	0.0	0.0	0.0	50.0
RiskValue_22	184630.0	0.0	0.03	0.0	0.0	0.0	0.0	2.0
RiskValue_23	184630.0	0.01	1.44	0.0	0.0	0.0	0.0	360.0
RiskValue_24	184625.0	5.61	2402.6	0.0	0.0	0.0	0.0	1032348.0
RiskValue_25	184625.0	0.01	1.44	0.0	0.0	0.0	0.0	360.0
RiskValue_26	184624.0	0.0	0.24	0.0	0.0	0.0	0.0	80.0
RiskValue_28	184625.0	0.0	0.12	0.0	0.0	0.0	0.0	7.0
RiskValue_29	184625.0	0.0	1.37	0.0	0.0	0.0	0.0	587.0
RiskValue_33	184625.0	inf		0.0	0.0	0.0	0.0	inf
RiskValue_34	184625.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
TRH_EquipmentChanges	754880.0	1.01	0.11	0.0	1.0	1.0	1.0	7.63
TRH_Prescribers	754873.0	1.0	0.11	0.0	1.0	1.0	1.0	4.0
Consumption	233018.0	22985.49	4360118.88	-14402.0	274.0	3121.0	7617.0	1052790826.0
Ratio	232957.0	7.5	5.69	-16.97	4.35	6.58	8.48	48.0

Taula 6.7: Estadístiques per columna de les dades.

A la taula 6.7 s'hi mostra l'anàlisi descriptiva per columnes del conjunt de dades.

Cal destacar que totes les variables a excepció de la variable predictiva són *continues*.

Pel que fa a l'**edat** del pacient (columna PatientAge) es pot observar que l'edat mitjana és d'uns 66 anys amb una desviació estàndard força elevada. Aquest fet indica que la majoria dels pacients són d'edat avançada.

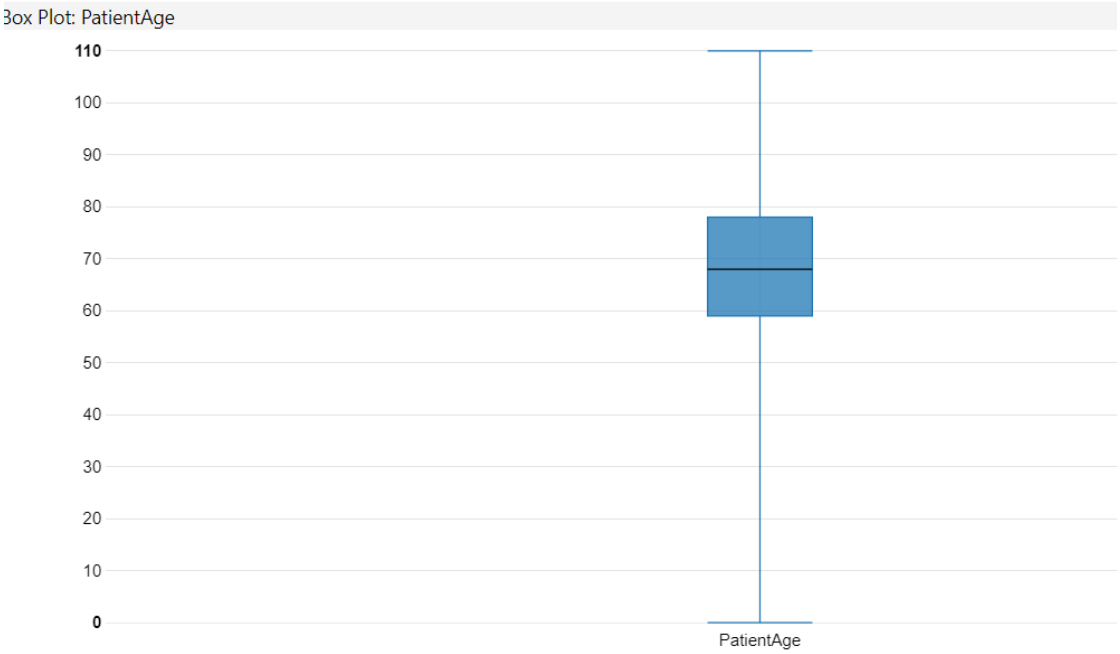

Figura 6.8: BoxPlot de la columna patientAge.

Pel que fa la setmana (columna WeekCreated) on la setmana 0 correspon al 16/7/2009 i la setmana 206 al 16/2013. Al següent boxplot es pot observar la dispersió de les dades en funció de la setmana de creació.

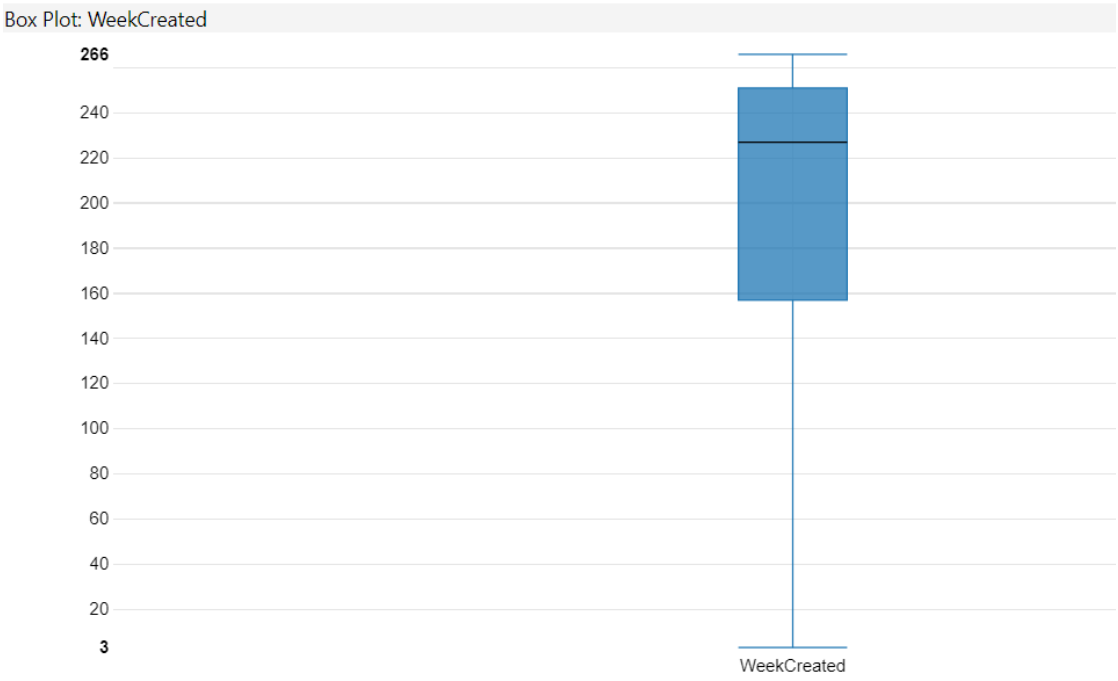

Figura 6.9: BoxPlot de la columna weekCreated.

Per altra banda, podem observar com en la major part dels casos, els pacients tenen una intervenció setmanal (sol ser la revisió que es fa setmanalment), però podem observar com el màxim és de 14 i en aquest cas trobem que hi ha un possible *outlier* ja que el segons els experts mèdics , no és habitual més de 3 intervencions setmanalment.

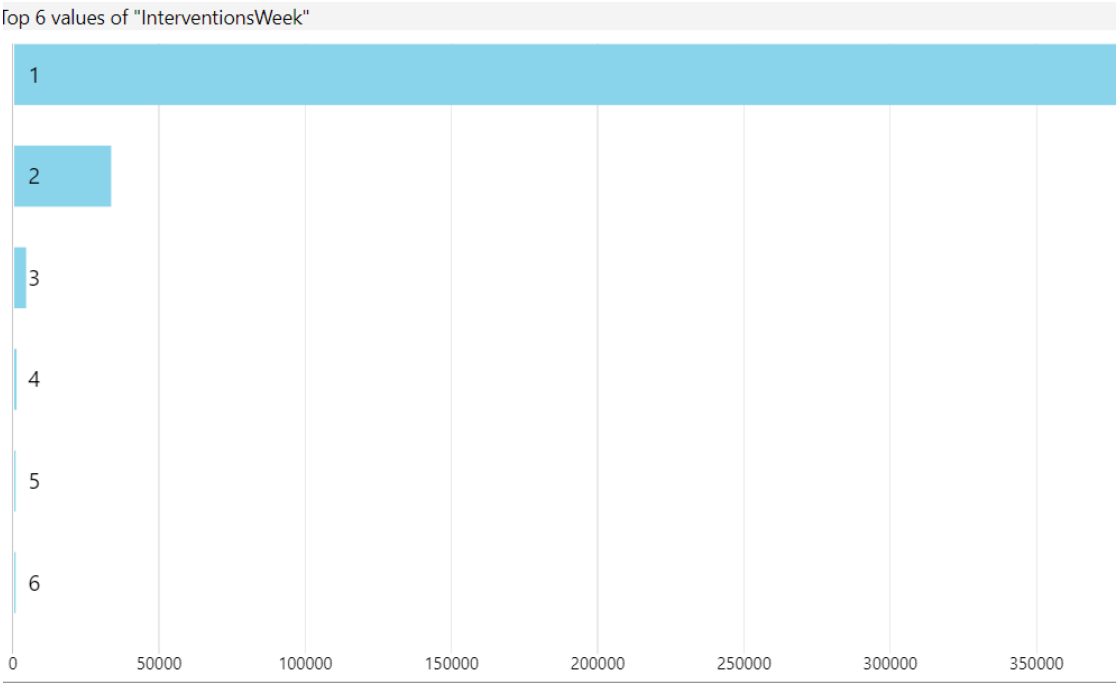

Figura 6.10: Valors mes freqüents del nombre d'intervencions setmanals per cada dupleta Pacient, Tractament

Pel que fa als **indicadors clínics** (columnes ClinicalValueX), deixant de banda la de tipus 6 ja que és la variable predictiva i serà analitzada independentment en seccions posteriors, cal destacar que pel que fa al indicador de tipus 15 veiem que el valor màxim és molt elevat respecte la mitjana i es pot tractar d'un valor anòmal (a la figura Per la resta d'indicadors clínics no es veu de forma aparent cap valor anòmal ni cap indicació de possibles problemes.

Pel que fa als **indicadors de progres** (columnes ProgressValueX) pel de tipus 2 veiem un valor mínim molt allunyat de la resta (veure boxplot). Pel de tipus 4 veiem un valor màxim molt elevat (veure boxplot) que també podria ser degut a alguna anomalia en les dades així com amb el cas de l'indicador de progrés de tipus 2.

Pel que fa als **indicadors de risc** (columnes RiskValuesX) el mes destacat que mostren les estadístiques és: en el cas de l'indicador de risc de tipus 5 veiem que el valor màxim està totalment allunyat de la resta, fet que també ens pot fer entendre que es tracta d'una anomalia. També es pot observar que el RiskValue de tipus 33 conté valors erronis. Aquest fet es deu a que no existeixen valors per a aquesta columna o bé que hi ha hagut algun problema a l'hora de traspassar les dades a la Base de Dades.
Finalment, observant les estadístiques per la columna **ratio** podem observar com el valor màxim i el mínim estan força allunyats però aquest fet no ens permet concloure que siguin dades anòmales ja que aquest valor segons els experts en els temes es poden donar

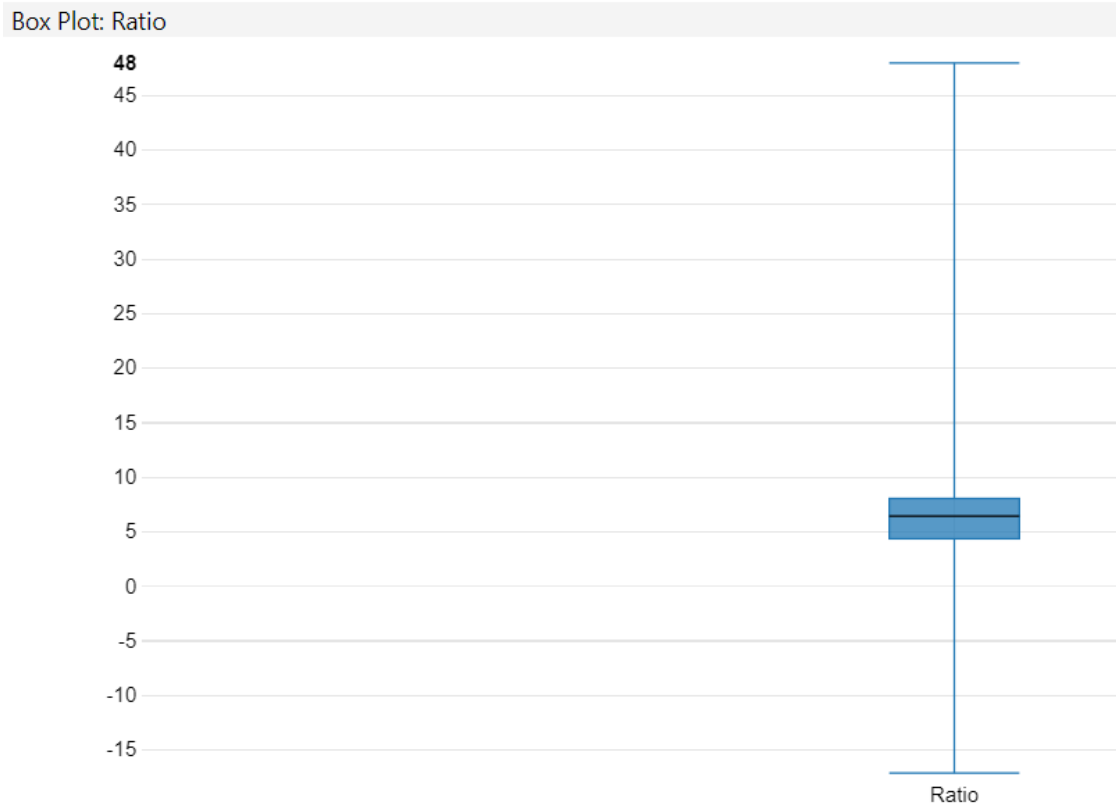

Figura 6.11: BoxPlot de la columna ratio

en casos puntuals com per exemple, episodis d'alt nivell de pol·len. A continuació es mostra un boxplot d'aquesta columna.

* Cal tenir en compte que en aquesta fase les dades ja no contenen valors nuls ja que aquests han estat eliminats en fases anteriors quan s'han eliminat les files en les quals existia algun valor nul procedent de l'agrupació de columnes.

6.4 Preprocessament i selecció de variables

6.4.1 Eliminació de valors anòmals

Abans de fer l'anàlisi de la variable predictiva, cal primer eliminar els valors anòmals que tenen les dades. L'objectiu és evitar que aquests tinguin una influència en la qualitat del model.
S'han eliminat els valors fora de rang de totes les variables indicadores, ja siguin de risc, clíniques, de progrés, etc. Ja que aquest són deguts a anomalies i podrien corrompre l'anàlisi i l'elaboració del model. Seguidament es mostren els resultats d'eliminar els

valors anòmals de les següents columnes [5]:

- **InterventionsWeek**: a la següent figura podem observar les estadístiques de la columna. Podem veure que el valor màxim és 6 i la majoria de valors entre el quantil 25 i el quantil 75 tenen valor 1. Aquest fet indica que la majoria de pacients tenen una intervenció setmanal per tractament. S'han eliminat els valors majors que 7.0.

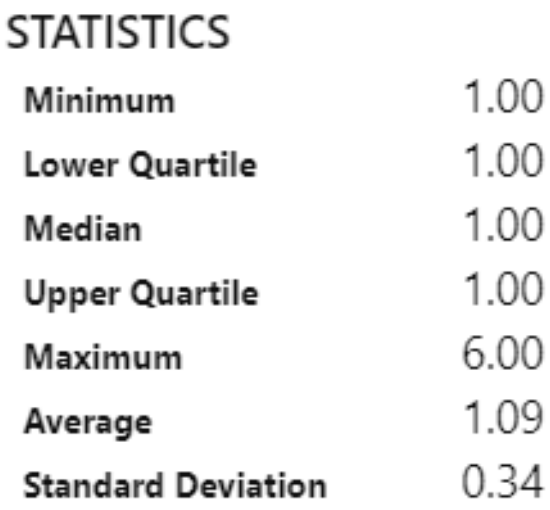

Figura 6.12: Estadístiques de la columna InterventionsWeek un cop eliminats els outliers

- **ClinicalValue8**: S'han eliminat els valors mes grans que 60 ja que consumir mes de 60l/min és molt improbable i segurament es tracta d'un error en la introducció de les dades. A la figura **??** es pot observar el boxplot després d'aplicar els canvis.

- **ClinicalValue15**: s'han eliminat els valors majors que 100. A la figura **??** es pot observar el boxplot després d'aplicar els canvis.

- **ProgressValue2**: s'han elminat valors que estan fora del rang = [-100,200]. Al codi de la figura **??** es pot observar el boxplot després d'aplicar els cambis.

- **ProgressValue4**: s'han eliminat els valors que estan fora del rang = [0,60]. A la figura **??** es pot observar el boxplot després d'aplicar els canvis.

- **RiskValue5**: s'han eliminat els valors fora del rang = [0,60]. A la figura **??** es pot observar el boxplot després d'aplicar els canvis.

[5] Es mostren només alguns exemples degut a la gran quantitat de columnes.

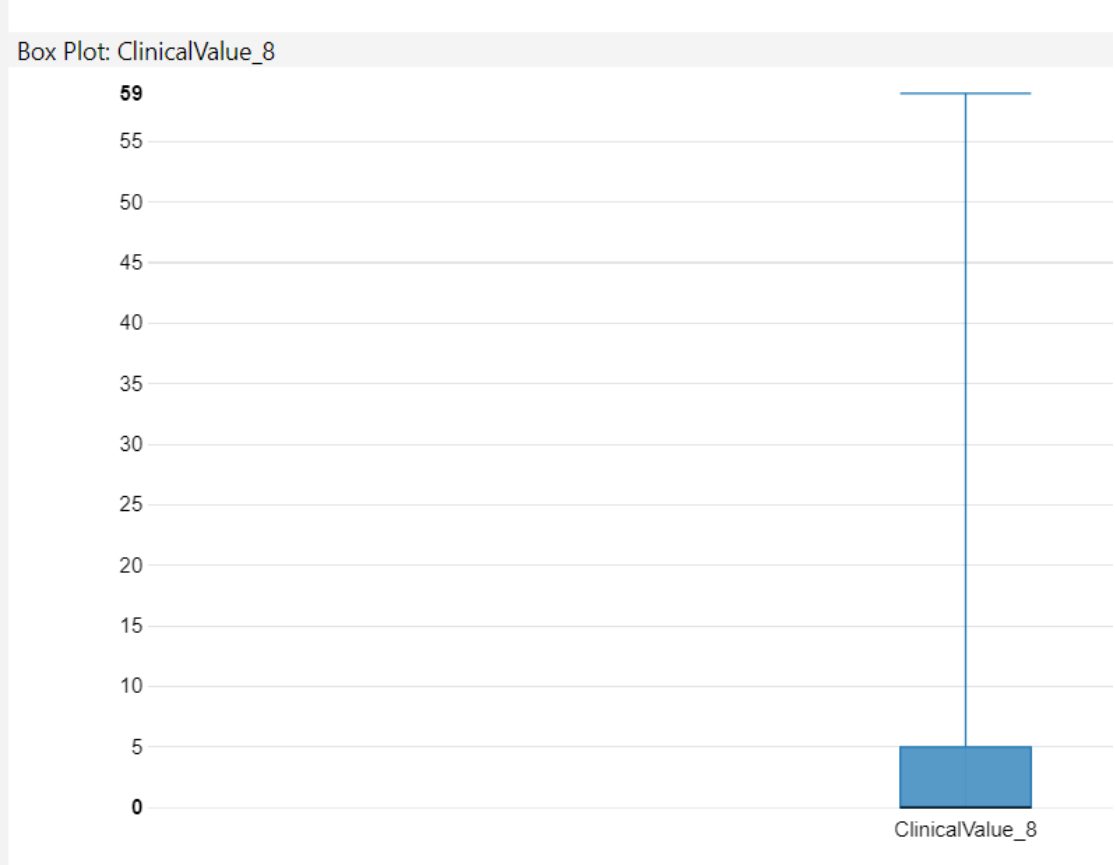

Figura 6.13: BoxPlot de la columna clinicalValue8 un cop eliminats els outliers

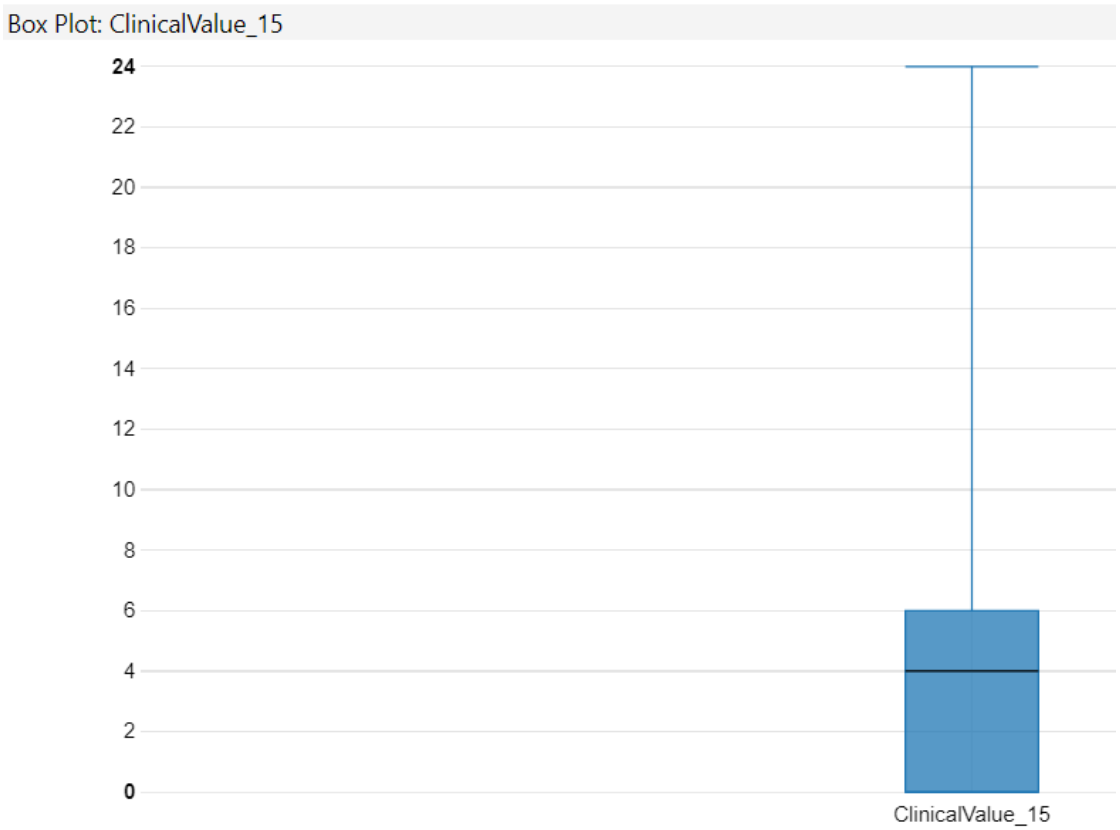

Figura 6.14: BoxPlot de la columna clinicalValue15 un cop eliminats els outliers

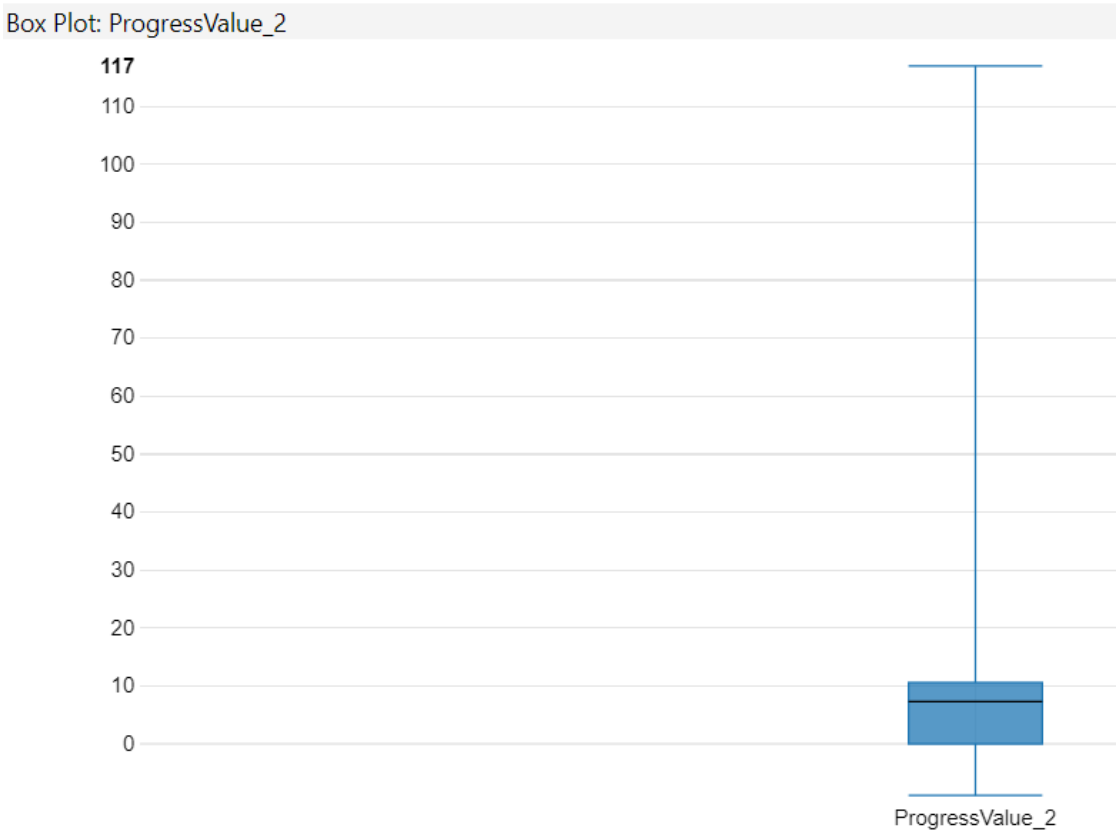

Figura 6.15: BoxPlot de la columna ProgressValue2 un cop eliminats els outliers

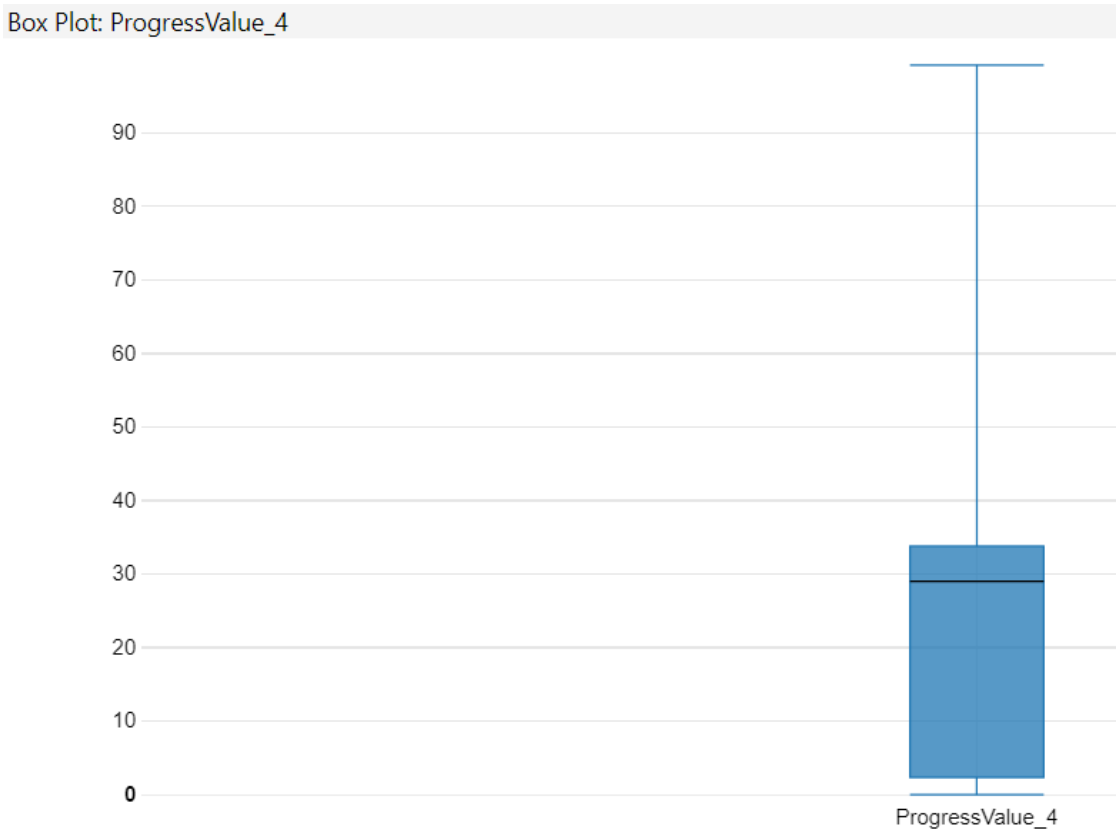

Figura 6.16: BoxPlot de la columna ProgressValue4 un cop eliminats els outliers

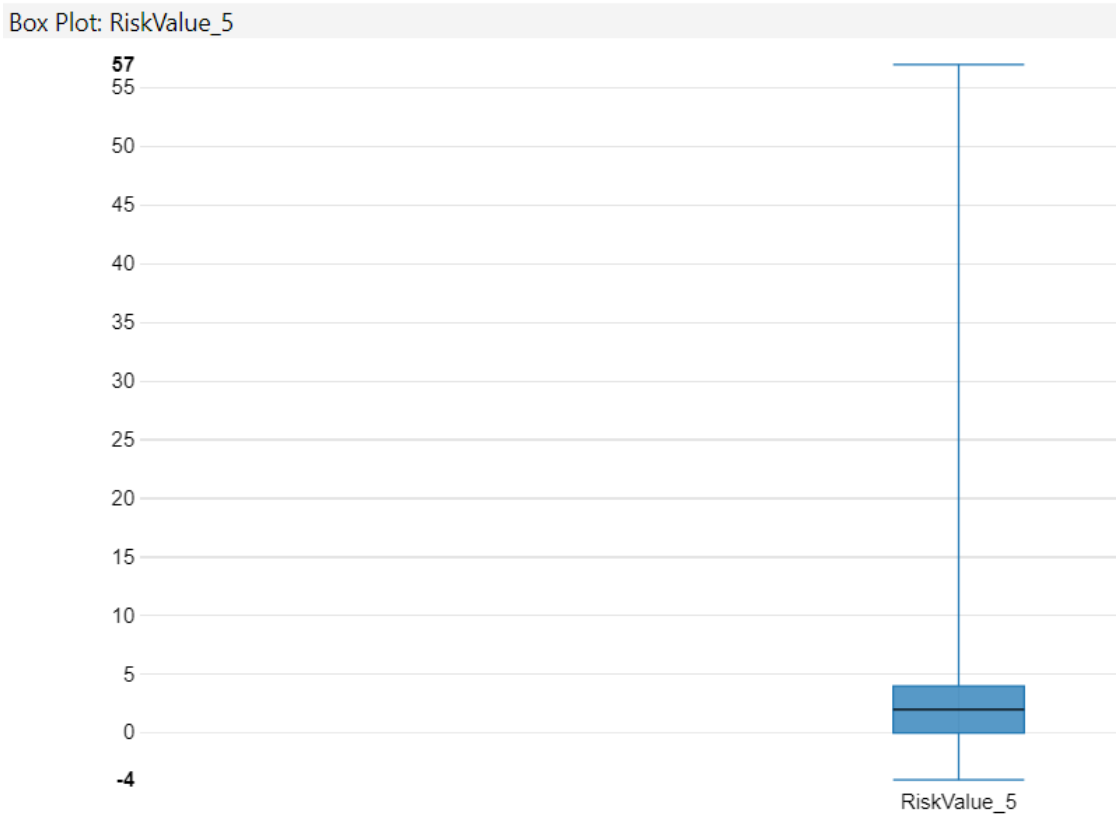

Figura 6.17: BoxPlot de la columna RiskValue5 un cop eliminats els outliers

6.4.2 Eliminació de variables redundants

Seguidament s'ha cercat si existeixen variables (diferents de la variable predictiva) que són redundants o molt similars entre elles. L'objectiu és eliminar variables que que estiguin correlacionades entre elles un 0.9% o mes. A la següent figura es pot veure el codi que s'ha utilitzat.

```
def find_correlation(data, threshold=0.9):
    corr_mat = data.corr()
    corr_mat.loc[:, :] = np.tril(corr_mat, k=-1)
    already_in = set()
    result = []
    for col in corr_mat:
        perfect_corr = corr_mat[col][corr_mat[col] > threshold].index.tolist()
        if perfect_corr and col not in already_in:
            already_in.update(set(perfect_corr))
            perfect_corr.append(col)
            result.append(perfect_corr)
    select_nested = [f[1:] for f in result]
    select_flat = [i for j in select_nested for i in j]
    return select_flat
```

El codi anterior fa el següent:

- **Paràmetres**:
  - **Data**: pandas DataFrame que conté la matriu de dades.
  - **threshold**: decimal. Valor de correlació a partir del qual s'elimina una de les variables de la parella.
- Cerca totes les parelles de columnes. Calcula la correlació entre cada parella i si és major que *threshold*, aleshores elimina la primera variable de la parella.
- Retorna el nom de les columnes a eliminar.

Un cop executat el codi, les variables a eliminar són les següents:

- ClinicalValue5
- ClinicalValue7
- ClinicalValue12
- ClinicalValue14
- ClinicalValue15
- ProgressValue15
- ProgressValue27

- ProgressValue45
- RiskValue12
- RiskValue15
- RiskValue18
- RiskValue24
- RiskValue26
- RiskValue29
- RiskValue34

El nombre total de columnes un cop aplicat aquest pas és de: 74.

6.4.3 Selecció de les característiques més rellevants

Malgrat que ja s'ha reduït força el conjunt de dades, el nombre de columnes segueix sent elevat. Observant els resultats de les seccions anteriors és plausible pensar que es pugui reduir aquest conjunt. L'objectiu d'aquesta secció doncs és trobar el nombre optim de columnes o característiques.

La selecció de categories rellevants o *Feature selection* [12] és el procés de seleccionar les característiques més rellevants per a utilitzar a la construcció del model. Els principals usos d'aquestes tècniques són:

- Simplificar els models per a una senzilla interpretació.
- Temps d'entrenament més curts.
- Reducció de la variància i la possibilitat d'un menor *overfitting*.

Pel cas que ocupa aquest treball s'ha decidit aplicar aquesta tècnica per la raó de simplificar els models i també per obtenir temps d'entrenaments més curts. La raó de reduir la variància també és interessant però més secundària pels objectius d'aquest treball.

Hi ha diferents tècniques possibles a l'hora d'aplicar la selecció de característiques com per exemple: *Univariate feature selection*, *recursive feature selection*, extreure les variables amb menys variança, etc.
En aquest treball s'ha decidit aplicar la tècnica de **recursive feature selection** [13]. Aquesta tècnica consisteix a assignar pesos a les variables i seleccionar aquestes considerant recursivament petits subconjunts de característiques. Primer l'estimador és entrenat amb el conjunt inicial de variables i calcula la importància de forma recursiva i elimina les menys influents segons el nombre desitjat de característiques.
Per a realitzar aquest procés s'ha utilitzat la lliberia *sci-kit learn* amb el llenguatge

python. En concret, aquesta llibreria té un paquet anomenat *sklearn feature_selection* que ens permet implementar aquesta funcionalitat: A continuació s'explica com s'ha realitzat el procés:

1. Primer s'ha d'inicialitzar un estimador. En aquest cas s'ha seleccionat un de tipus *RandomForestClassifier* ja que és un estimador que generalitza bé en general.

```
clfRf4 = RandomForestClassifier()
```

2. Seguidament s'inicialitza l'objecte RFECV que inicialitza l'estimador. A més a més. S'ha decidit utilitzar *20-fold Cross-Validation* per cerciorar-nos de que l'estimador generalitza de forma correcta. La mètrica per tal de discernir entre la millor variable que s'utilitza és l'encert (*Accurracy*).

```
rfecv = RFECV(estimator=clf_rf_4 , step=1, cv=20,scoring='accuracy')
```

3. Finalment s'entrena el model per tal de trobar el nombre òptim de característiques.

Un cop aplicats els passos el resultat obtingut és el següent:

- Nombre optim de característiques: 13.

- Característiques més rellevants: PatientId, TreatmentId, PatientAge, WeekCreated, ClinicalValue8, ClienicalValue9, ClinicalValue10, ClinicalValue11, ClinicalValue13, ClinicalValue15, ClinicalValue18, ClinicalValue19, i PrescriptorChanges.

A partir dels resultats és interessant destacar que les variables més importants fan referència als indicadors clínics. També es pot veure com el pacient i el tractament en qüestió influeixen significativament sobre la variable predictiva així com el ratio.

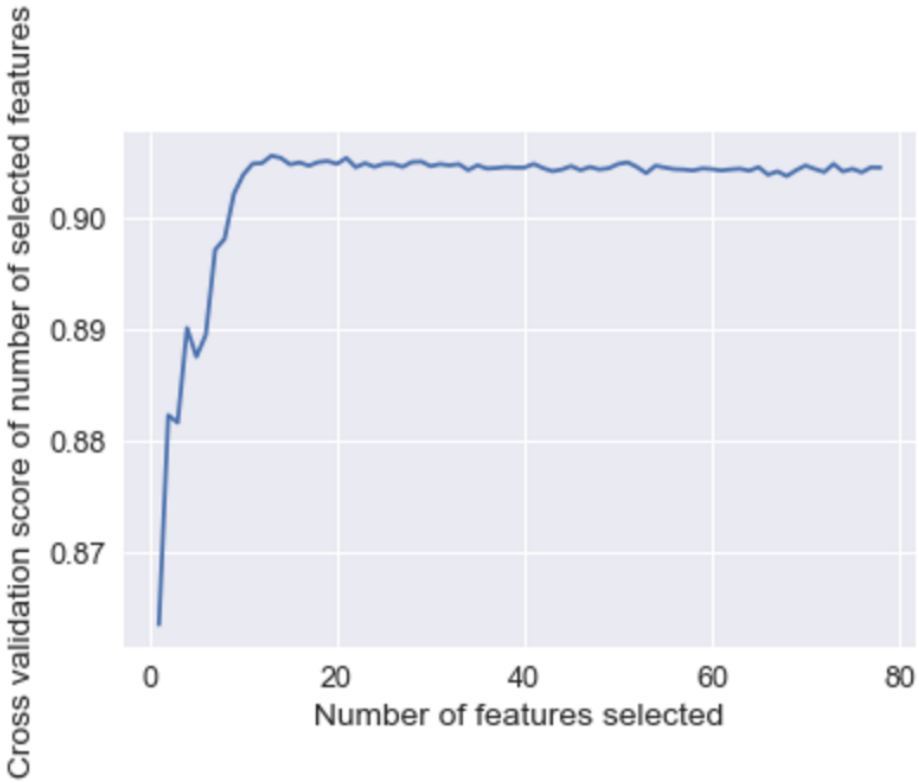

Figura 6.18: Nombre de característiques seleccionades (X) vs CV-Score(Y)

A la figura 6.18 es pot veure la relació entre el nombre de característiques i l'encert que s'obté a partir d'aquestes. De la gràfica en podem extreure la conclusió que a partir de les 10 característiques l'encert no varia massa i, per tant, no s'obtindria a priori un millor model entrenant-lo amb més de 10 variables.

Als resultats es pot observar que les variables fixes (PatientId, TreatmentId, PatientAge). Aquest fet pot ser degut a que aquestes variables es repeteixen en el conjunt de dades degut a com s'ha plantejat el problema. En futures versions es pot fer un re-anàlisi del conjunt de dades per tal de veure si realment aquestes variables han de ser rellevants o no.

6.5 Anàlisi de la variable predictiva

Un cop aplicades les fases de preprocessat de les dades i com a pas previ al desenvolupament dels models predictius, es mostrarà un anàlisi de la variable predictiva. S'ha realitzat un anàlisi individual sobre la variable i també la influència que tenen la resta de variables seleccionades sobre aquesta.

La variable predictiva és l'indicador clínic de tipus 6. Aquest element indica el nombre de dies setmanals en el que el pacient s'ha posat la màscara d'oxigen més de 3 hores.

El client ha mostrat interès en aplicar el focus en aquesta variable ja que és una variable que el valor i la implicació d'aquest es pot entendre pel pacient de forma senzilla i sense haver de requerir coneixements mèdics ni tècnics. Té sentit doncs escollir aquesta variable ja que l'aplicació està enfocada als pacients.

Com ja es pot intuir els valors que pot prendre aquesta variable estan compresos entre 0 (cap dia de la setmana ha estat necessari dur la màscara més de 3 hores) fins a 7 (el pacient ha dut la màscara més de 3 hores tots els dies de la setmana).

El fet de tenir un rang de valors **discret** fa que el problema a abordar es pugui plantejar com un problema de **classificació** com es detallarà i justificarà en seccions posteriors.

6.5.0.1 Anàlisi individual

Un cop explicat el significat de la variable i les implicacions que comporta, a continuació es mostren tan les estadístiques individuals d'aquesta així com gràfiques que ens permeten entendre el contingut d'aquesta:

STATISTICS

Minimum	0.00
Lower Quartile	3.00
Median	6.00
Upper Quartile	7.00
Maximum	7.00
Average	4.81
Standard Deviation	2.82

Figura 6.19: Estadístiques de la variable indicativa

Observant les estadístiques i les gràfiques a les figures **??**, **??** i **??** es pot observar que les dades **no** estan **balancejades**, és a dir, hi ha moltes més observacions d'unes categories que de la resta: hi ha moltes més observacions de 0 i 7 dies que de la resta. Aquest fet implica, per exemple, que malgrat l' encert sigui elevat el model s'està focalitzant en les classes amb més ocurrències.

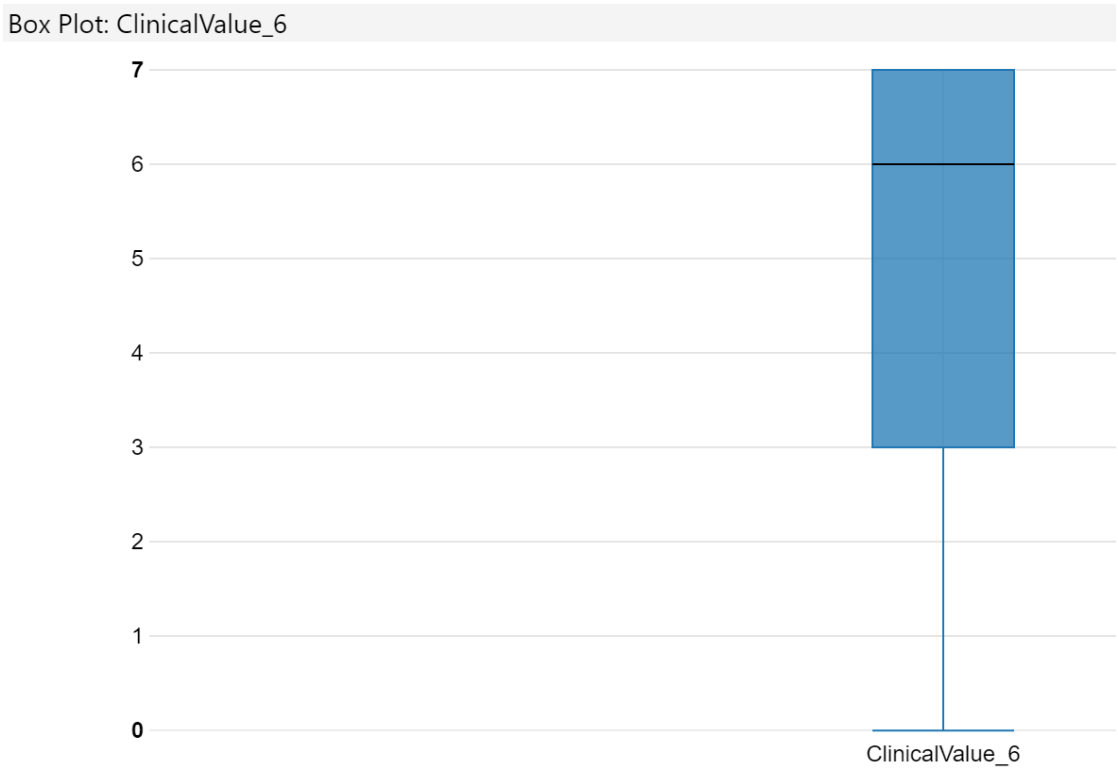

Figura 6.20: Boxplot de la variable indicativa

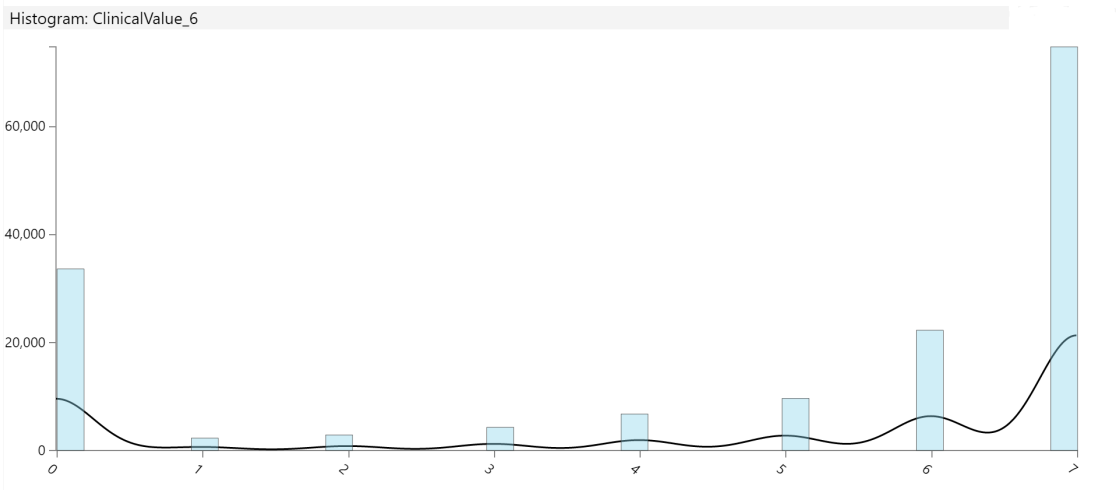

Figura 6.21: Histograma de la variable indicativa.

Un altre conclusió que podem extreure és que la desviació estàndard és de ±2.82. Un valor elevat tenint en compte que el rang de valors és = [0,7]. Aquest fet segurament és degut al no balanceig de les dades, però és important mencionar-ho i tenir-ho en compte.

6.5.0.2 Anàlisi respecte la resta de variables

La figura 6.22 mostra la matriu de correlació amb les 15 columnes més correlacionades amb la variable indicativa (ordenades de major a menor). A simple vista, es pot veure que les variables de tipus indicadors ja siguin clínics o bé de progrés són les mes correlacionades amb la variable.

6.6 Construcció, avaluació i desplegament del model

En aquesta secció s'explicaran els passos seguits per tal de desenvolupar, avaluar i construir un model predictiu. L'*objectiu* d'aquesta fase és construir un model apte per ser utilitzat per qualsevol aplicació.

6.6.1 Infraestructura

Abans d'explicar com s'han desenvolupat les fases, cal conèixer la infraestructura que s'ha utilitzat. Com ja s'ha dit anteriorment, cal una infraestructura per poder utilitzar el model creat en qualssevol aplicació.

Una manera d'aconseguir aquest objectiu és treballar al **núvol**. S'ha decidit utilitzar la plataforma Azure com a base per desenvolupar aquesta fase.
Azure [14] és un plataforma al núvol que ofereix gran quantitat de serveis i APIs (*Aplication programming Interfaces*) que poden ser utilitzades pels desenvolupadors tan per allotjar les seves aplicacions com també per utilitzar els diferents que ofereix.

Un dels diferents serveis que ofereix són els anomenats *Azure Machine Learning Services* [15]. Aquest conjunt de serveis permeten construir, desplegar i gestionar models usant les principals biblioteques i utilitats de Python. A més a més, també brinda eines per tal de poder desplegar els models creats al núvol i posar operatives API's REST que puguin ser utilitzades des de qualsevol tipus d'aplicació independentment de l'arquitectura.

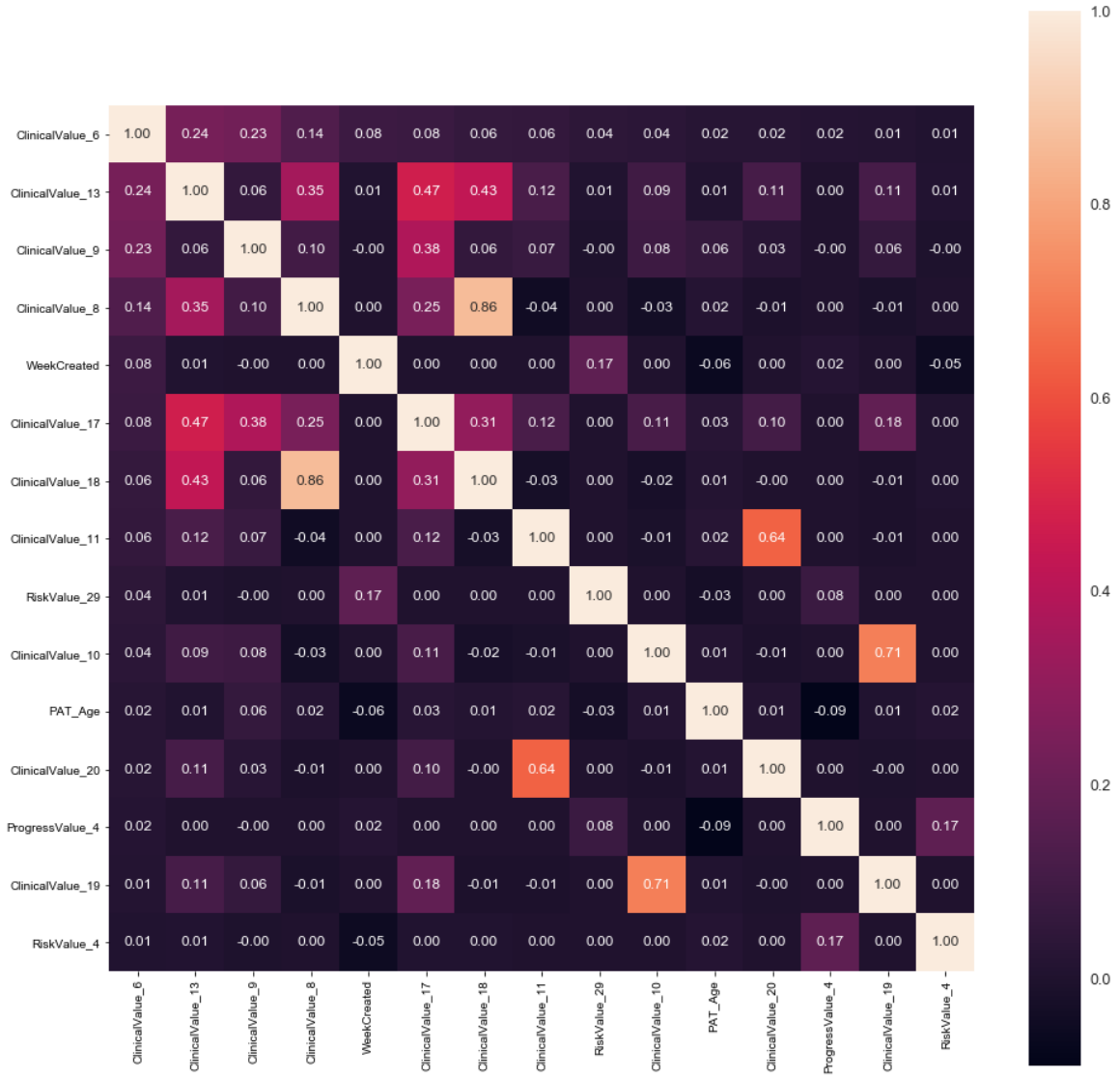

Figura 6.22: Matriu de correlació amb les 10 columnes més correlacionades amb la variable indicativa.

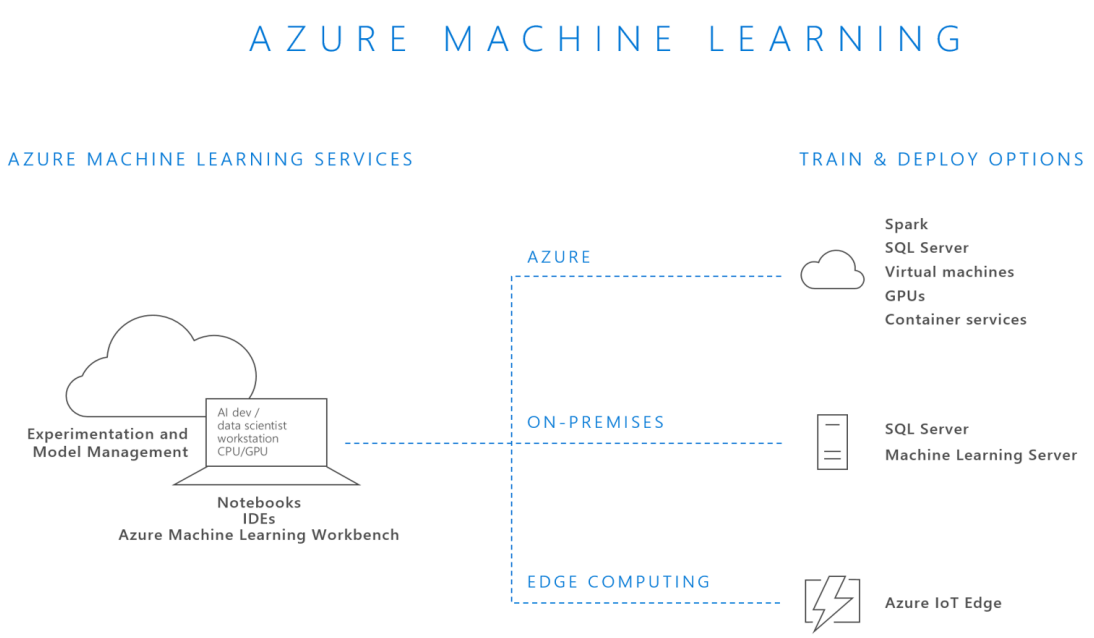

Figura 6.23: Esquema de la infraestructura Azure Machine Learning Services. Imatge extreta de Microsoft Azure Documentation [1].

A la figura 6.23 es pot veure un esquema de la infraestructura que ofereixen aquests serveis. Per a aquest projecte, les part *on-premises* i *edge-computing* no s'utilitzaran malgrat que és interessant tenir-les en compte per futures extensions del treball o fins i tot treballs alternatius relacionats amb aquest.

La principal eina que ofereix aquest servei per tal de desenvolupar i desplegar els models al núvol és l'anomenada **Azure Machine Learning Workbench** [16]. Es tracta d'una aplicació d'escriptori a més d'un seguit seguit d'eines per línia de comandes suportada tant en sistemes Windows com MacOS que permet gestionar solucions basades en *Machine Learning*, des de l'anàlisi de dades fins al desplegament de models. Aquesta eina ofereix multitud de funcionalitats però a continuació es destaquen les que s'han utilitzat per la realització d'aquest projecte [6]:

- Desenvolupament i experimentació amb models: l'eina porta integrada les principals biblioteques en Python com per exemple *sci-kit learn* [17].Sci-kit learn és una biblioteca de codi obert que ofereix gran varietat d'algoritmes de Machine Learning i altres eines relacionades. Aquesta llibreria és la principal llibreria utilitzada pel desenvolupament dels models.

- Desplegament de models: ofereix eines per línia de comandes per tal de desplegar

[6] Aquesta eina també proporciona recursos per preprocessar i transformar les dades. No obstant, aquests no s'han utilitzat ja que en el moment de la realització de la fase, aquestes eines encara es troben en una fase preliminar i s'ha decidit no utilitzar-les degut a aquest motiu.

un servei al núvol a partir d'un fitxer que conté la informació del model. Aquest servei s'utilitza per obtenir una o varies prediccions a partir d'un conjunt de dades (fila) donat.

6.6.2 Desenvolupament i avaluació del model

A continuació es descriu com s'ha desenvolupat el model. L'objectiu d'aquesta fase és aconseguir un model apte per ser desplegat.

Per a desenvolupar el model s'han provat 3 mètodes diferents: RandomForest, Multi-Class Support Vector Machine i AdaBoost. S'han escollit aquests degut a que són mètodes ja que no són ni tan simples com la regressió multinomial ni tan sofisticats com altres mètodes relacionats amb el *Deep Learning* els quals requereixen coneixements força avançats que queden fora de l'abast dels objectius d'aquest projecte. També s'ha de tenir en compte que aquest projecte té diverses fases i focalitzar-se molt en aquesta fase podria haver-ne alterat la planificació.

Per altra banda, es podria haver optat per seleccionar més d'un model i desplegar-lo al núvol. No obstant, s'ha de tenir en compte que desplegar serveis al núvol té un cost econòmic. Com que els usuaris finals seran els pacients, desplegar i utilitzar més d'un model a l'aplicació no aportaria un valor extra com a funcionalitat.

Cal recordar que el problema que volem afrontar és de **classificació** amb múltiples classes. L'objectiu es trobar un model que classifiqui el nombre de dies setmanals que el pacient durà una màscara d'oxigen durant més de 3 hores. La variable predictiva és doncs l'indicador clínic de tipus 6 (variable ClinicalValue_6) del conjunt de dades.

Degut a que els possibles valors de la variable predictiva són ordinals en en rang enter [0,7] també es pot plantejar el problema com un problema de **regressió**. El model podria predir valors decimals i no enters. En un principi aquest fet no seria un problema degut a que es podria arrodonir el resultat a l'enter més proper. No obstant es pot donar el cas on el model retorni prediccions allunyades dels possibles valors. Aquest fet és el principal motiu pel qual s'ha descartat aquesta opció. Com es pot comprovar a la secció 6.6.2.5 els resultats no són suficientment bons comparats amb els mètodes de classificació. És interessant però explorar aquesta possibilitat ja que en possibles extensions o millores podria ser un element a tenir en compte.

6.6.2.1 Descripció dels experiments

Per a dur a terme l'avaluació per cadascun dels mètodes seleccionats: AdaBoost Classifier, RandomForests i Multi-class Support Vector Machine, a continuació es descriu l'experiment comú per cadascun dels 3 mètodes[7]:

[7] Aquest passos només s'apliquen pels models de classificació. Pel model de tipus regressió es farà una experimentació similar que s'explica a la secció 6.6.2.5

1. Es divideixen amb dos conjunts:
   - Dades d'entrenament: 80%.
   - Dades de test: 20%.
2. S'executa l'entrenament utilitzant 20-fold CV amb el conjunt d'entrenament.
3. Es calcula la **matriu de confusió** amb les dades d'entrenament.
4. Es calcula l'**error d'entrenament**.
5. Es calcula la **matriu de confusió** amb les dades de test.
6. Es calcula l'**error de test**.

6.6.2.2 AdaBoost Classifier

6.6.2.2.1 Descripció del mètode

AdaBoost [18] és un *meta-algoritme* que utilitza tècniques de *Boosting* [19]. Aquestes tècniques consisteixen a combinar diferents algoritmes normalment més simples com els **arbres de decisió** [20]. El resultat d'aplicar aquests algoritmes és combinat amb una suma ponderada que representa el resultat final del classificador AdaBoost.

Per al nostre experiment s'utilitzarà com a algoritmes d'aprenentatge un conjunt d'arbres de decisió.

Els principals hiper-paràmetres de l'algoritme són:

- **Estimador base**: estimador base amb el qual es construeix el mètode d'ensamblatge. A la llibreria Sci-kit learn el valor per defecte és : Arbres de Decisió.
- **Nombre d'estimadors**: màxim nombre d'estimadors al qual el mètode acaba. Si el mètode convergeix abans d'arribar al màxim, llavors l'algoritme acaba. A la llibreria Sci-kit learn el valor per defecte és : 50 estimadors.
- **Alogitme d'ensamblatge**: algoritme d'optimització. A la llibreria Sci-kit learn el valor per defecte és : SAMME.R.

6.6.2.2.2 Execució i avaluació del mètode

A continuació s'exposen els resultats d'entrenar i avaluar el mètode **AdaBoost**. El procediment pe a realitzar l'experiment és el descrit a la secció 6.6.2.1.

Els diferents valors a optimitzar pels paràmetres de l'algoritme són els següents:

- **Estimador base**: Arbre de decisió amb 5 nivells màxims de profunditat.
- **Nombre d'estimadors**: tots els valors entre el rang [0,300].

- **Alogitme d'ensamblatge**: SAMME.R

El primer pas és entrenar el model utilitzant la tècnica de *cross-validation* tal com s'ha comentat a la secció 6.6.2.1. L'objectiu és trobar el nombre òptim d'estimadors.

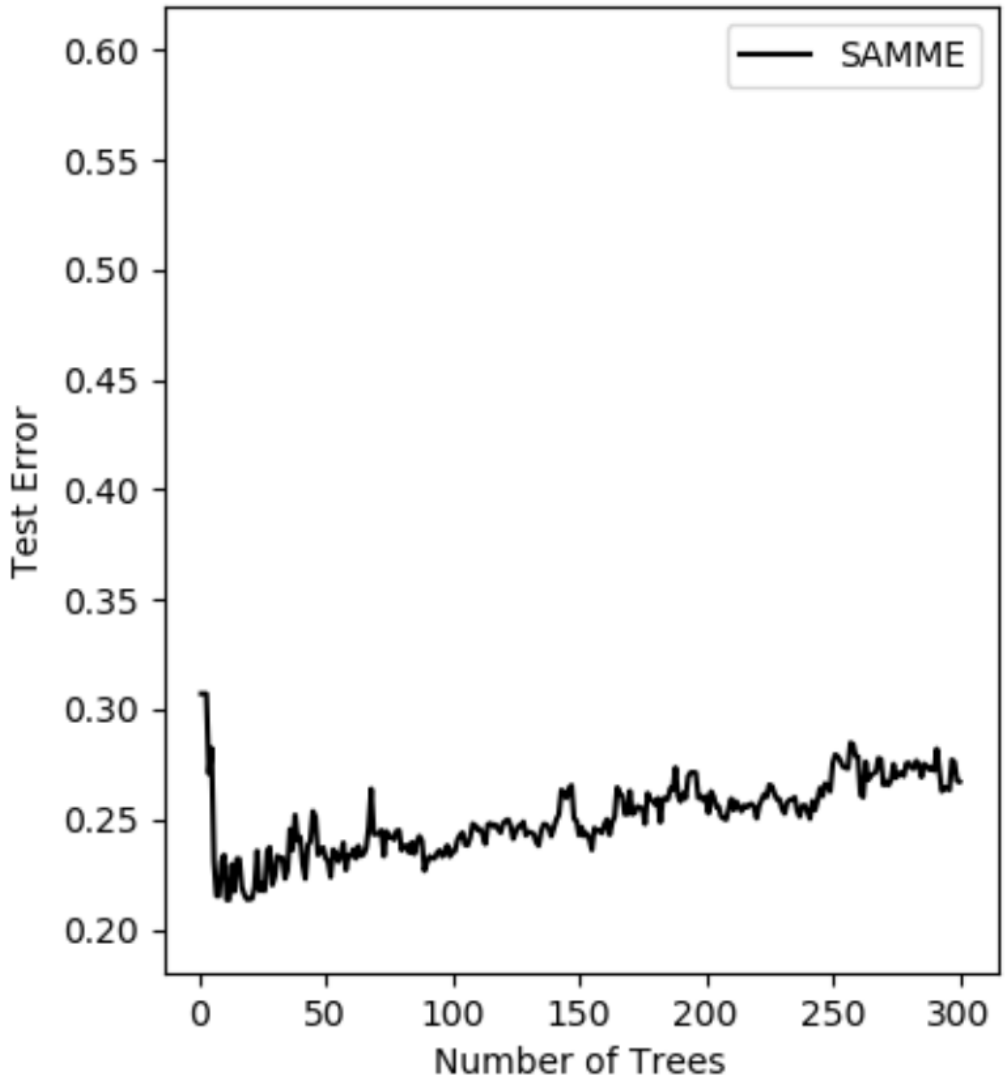

Figura 6.24: Gràfica de l'error en funció del nombre d'estimadors pel model AdaBoost

Un cop aplicat el procediment, el nombre òptim d'estimadors és de 20 estimadors. A la gràfica de la figura 6.24 es pot observar l'evolució en funció del nombre d'estimadors (arbres de decisió). La tendència de l'error tendeix a créixer aproximadament a partir

dels 50 estimadors.

Un cop seleccionats el nombre òptim d'estimadors, el pas següent és entrenar el model amb aquest valor. S'ha calculat l'encert tant amb les dades de **training** com amb les de **test**. Els resultats es mostren a la següent taula:

	Encert (%)
Training	99.17
Test	92.12

Taula 6.8: Encert (training vs test) del model AdaBoost

Es pot observar que en ambdós casos l'encert està per sobre del 90%. Pels resultats es pot observar com hi ha un possible *overfitting* ja que l'error de training és sensiblement més elevat que el de test.

Per observar a com classifica el model, a continuació es mostren les matrius de confusió tant amb les dades de test com amb les dades d'entrenament:

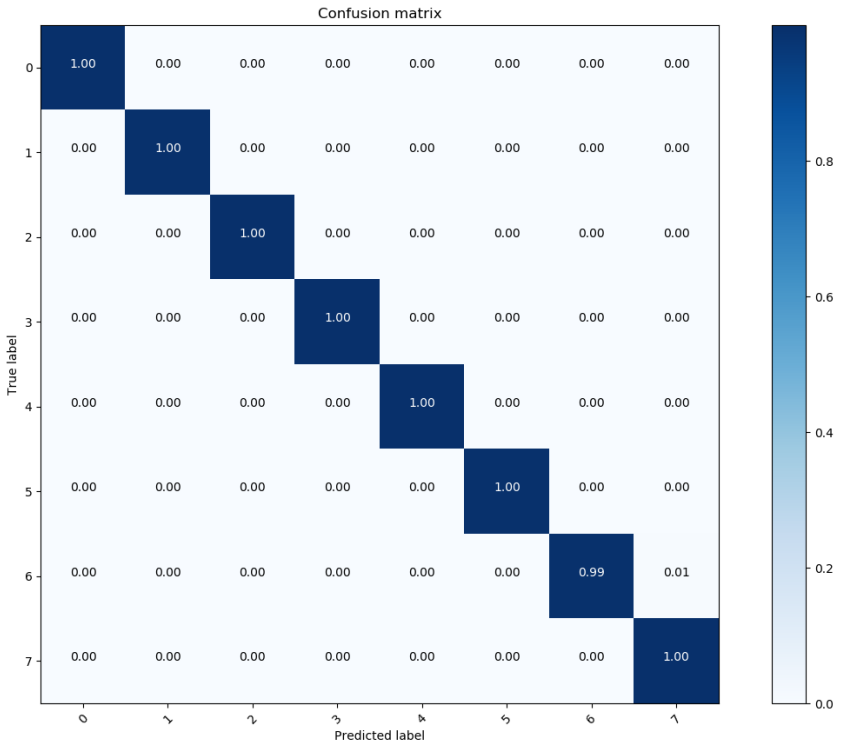

Figura 6.25: Matriu de confusió amb les dades d'entrenament del model AdaBoost

Figura 6.26: Matriu de confusió amb les dades de test del model AdaBoost

A les figures 6.29 i 6.30 es poden veure les matrius de confusió resultat de predir el model amb les dades d'entrenament i de test respectivament. En ambdós casos es pot observar com el comportament és molt similar: per les classes amb més observacions (0 i 7 dies) el model té una tassa d'encert més elevada que la resta de classes amb menys observacions. S'observa una tendència clara a la predicció de les classes majoritàries. Una altra conclusió que es pot extreure és que malgrat en general, l'error de test és més elevat que el de training, el comportament és molt similar en els dos casos.

Per altra banda, un element a tenir en compte en problemes de classificació és el "trade-off"entre la *precision* i el *recall*. A la figura 6.27 es pot observar la gràfica precision contra recall. Es pot veure com el valor de la recall (valors positius predits correctament) la precisió disminueix fins al 40% aproximadament.

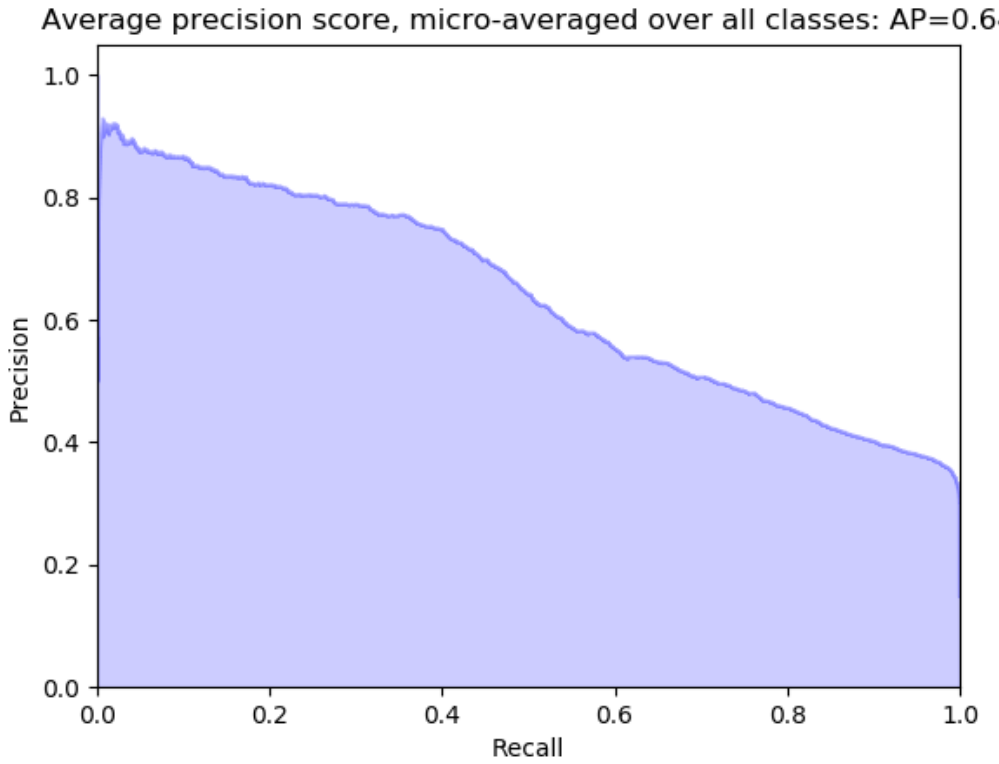

Figura 6.27: Gràfica precision vs recall pel model Adaboost

6.6.2.3 Random Forests

6.6.2.3.1 Descripció del mètode

Random forests [21] és un mètode d'ensamblatge. La diferència amb el mètode AdaBoost és que en comptes de que cada algoritme analitzi totes les variables i llavors es ponderi el resultat, el que es fa és que cada mètode selecciona aleatòriament un conjunt de variables i el resultat és la suma ponderada dels submètodes. Una altra diferència és que els submètodes només poden ser arbres de decisió i no qualssevol com en el cas del mètode AdaBoost.

Els principals hiper-paràmetres de l'algoritme són:

- **Nombre d'estimadors**: màxim nombre d'estimadors al qual el mètode acaba. Si el mètode convergeix abans d'arribar al màxim, llavors l'algoritme acaba. A la llibreria Sci-kit learn el valor per defecte és : 10 estimadors.
- **Criteri**: funció que té per objectiu mesurar la qualitat de la divisió. En Sci-kit learn és valor per defecte és: gini.

- **Màxima profunditat de l'arbre**: màxima profunditat que pot tenir cadascun dels arbres de decisió que intervenen. En Sci-kit learn és per defecte no té un valor predefinit.

- **Nombre màxim de variables per estimador**: nombre màxim de característiques amb les que pot treballar cadascun dels estimadors.

6.6.2.3.2 Execució i avaluació del mètode

Els diferents valors a optimitzar pels paràmetres de l'algoritme són els següents:

- **Nombre d'estimadors**: tots els valors compresos en el rang [2,200].

- **Criteri**: gini.

- **Màxima profunditat de l'arbre**: s'ha provat a no posar límit al valor i a que el valor màxim sigui $\log_2 N$ on N és el nombre de característiques.

Un cop seleccionats els paràmetres i entrenar el model amb les dades d'entrenament i *20-fold Cross-Validation*, a continuació s'analitzen els resultats.

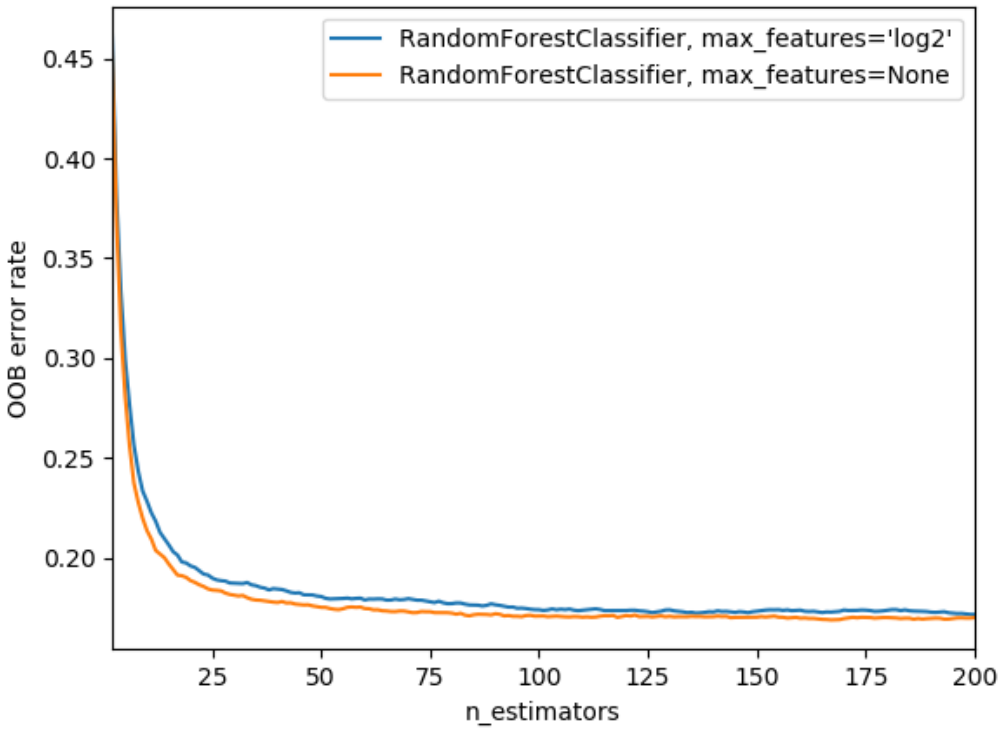

Figura 6.28: Nombre d'estimadors vs error per cada valor màxim en el nombre de variables de cada estimador

A la gràfica de la figure 6.28 es pot observar com varia l'error en funció del nombre d'estimadors per cadascun dels valors màxims de característiques per estimador. En primer lloc, es pot observar com el comportament és molt similar malgrat que sense posar límit l'error disminueix més ràpidament i el valor és mes òptim. Per tant, és millor no limitar el valor màxim de característiques per estimador. Per altra banda també es pot observar que el valor òptim d'estimadors és de 10.

Un cop seleccionats els paràmetres òptims, s'ha tornat a entrenar i provar el model escollint els valors òptims dels paràmetres. La següent taula mostra l'encert tant amb les dades de **training** com amb les de **test** un cop seleccionat els paràmetres òptims:

	Encert (%)
Training	82.54
Test	81.86

Taula 6.9: Encert (training vs test) del model RandomForest

Es pot observar que l'error l'encert amb les dades de test és aproximadament un 1% inferior a l'encert amb les dades d'entrenament. Les diferències no són molt notables i no es veu a priori que pugui existir *overfitting* No obstant, a continuació es mostren les respectives matrius de confusió per tal d'analitzar més en detall el comportament del mètode:

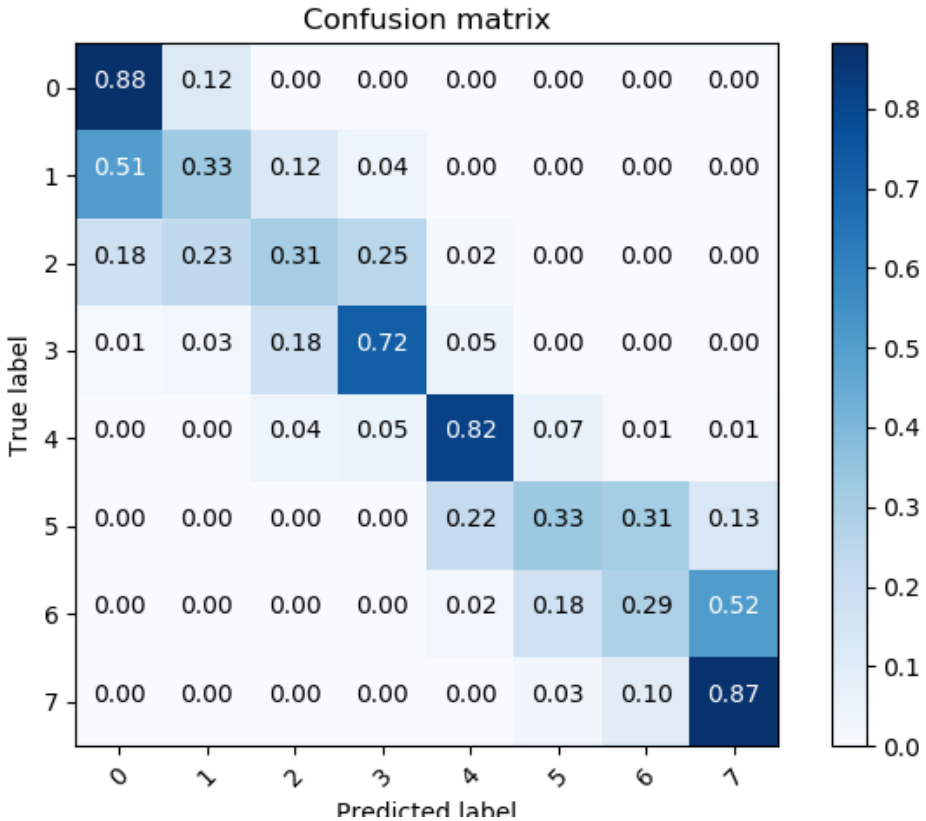

Figura 6.29: Matriu de confusió amb les dades d'entrenament del model RandomForests

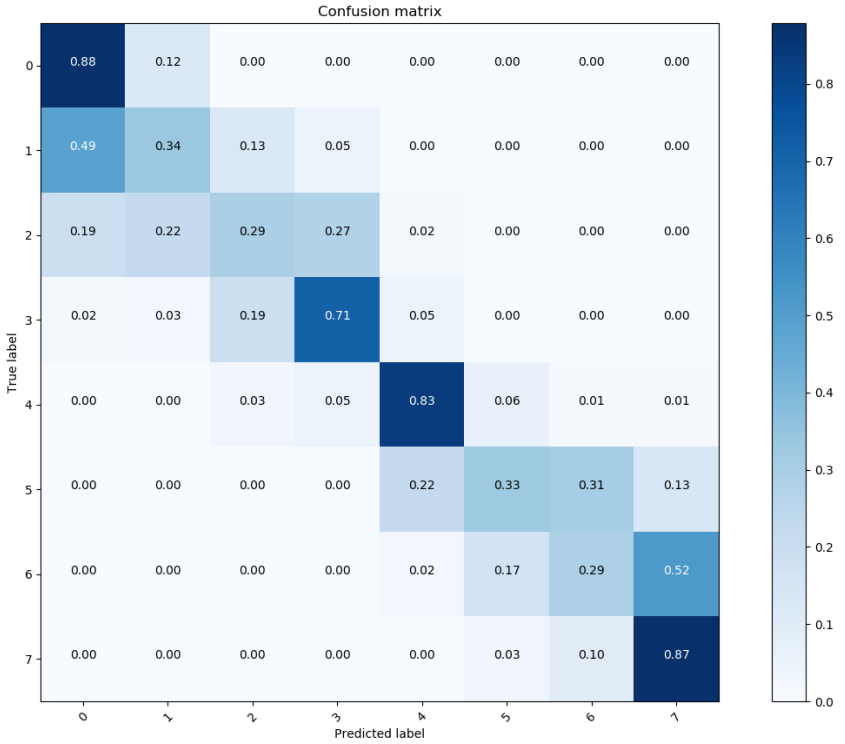

Figura 6.30: Matriu de confusió amb les dades de test del model RandomForests

Es pot observar una clara diferència entre el comportament de l'algoritme amb les dades de test i les d'entrenament. Pel que a la matriu resultant de la predicció amb les dades d'entrenament, podem observar que l'encert és quasi perfecta. Per altra banda, el comportament amb les dades de test és molt similar al que s'ha vist amb el cas de les dades de test del model AdaBoost. Podem concloure que el model amb gran probabilitat està sobreestimant les dades.

6.6.2.4 Multi-class Support Vector Machine

6.6.2.4.1 Descripció del mètode

SVM [22] és un mètode d'aprenentatge supervisat que pot ser utilitzat tant per problemes de classificació com de regressió. L'algoritme construeix un model que assigna nous exemples a una categoria o una altra, cosa que el converteix en un classificador lineal binari no probabilístic. Un model SVM és una representació dels exemples com a punts en l'espai, mapejats perquè els exemples de les categories separades estiguin el màxim de separades possible.

També es poden utilitzar per fer classificació entre múltiples classes (el cas que ocupa

aquest treball). Si tenim N classes, aleshores el mètode construeix N*(N-1)2 classificadors i cadascun entrena les dades amb 2 classes.

Els principals hiper-paràmetres de l'algoritme són:

- **C**: aquest paràmetre aplica un *trade-off* entre la complexitat i la proporció dels exemples no-separables. Una C amb un valor baix permet mes errors però també produeix un marge més alt. Quan C tendeix a infinit, aleshores la penalització per error és més elevada. En Sci-kit learn és valor per defecte és: 1.
- **kernel**: en sci-kit learn per defecte: RBF
- **Gamma**: complexitat del kernel. En Sci-kit learn el valor per defecte 1N on N és el nombre de columnes.

6.6.2.4.2 Execució i avaluació del mètode

Els diferents valors a optimitzar pels paràmetres de l'algoritme són els següents:

- **C**: 0.01, 0.1, 1 ,10, 100.
- **kernel**: RBF
- **Gamma**: 0.1, 1, 10, 100.

El primer objectiu és trobar el valor òptim dels paràmetres aplicant *20-fold CV* utilitzant el conjunt d'entrenament tal com s'explica a la secció 6.6.2.1. Els valors òptims dels paràmetres un cop finalitzada l'execució són els següents:

- **C**: 10.
- **kernel**: RBF
- **Gamma**: 100.

Un cop obtinguts els valors òptims pel paràmetres, s'ha entrenat el model amb les dades d'entrenament i s'ha fet la predicció tant amb aquestes com amb el conjunt d'entrenament i s'han obtingut el següents encerts:

	Encert (%)
Training	100
Test	48.32

Taula 6.10: Encert (training vs test) del model Multi-Class SVM

Com es pot observar, el model SVM tendeix al *overfitting* clarament els resultats ja que l'error de training és 0 mentre que l'error de test és de més del 50%.

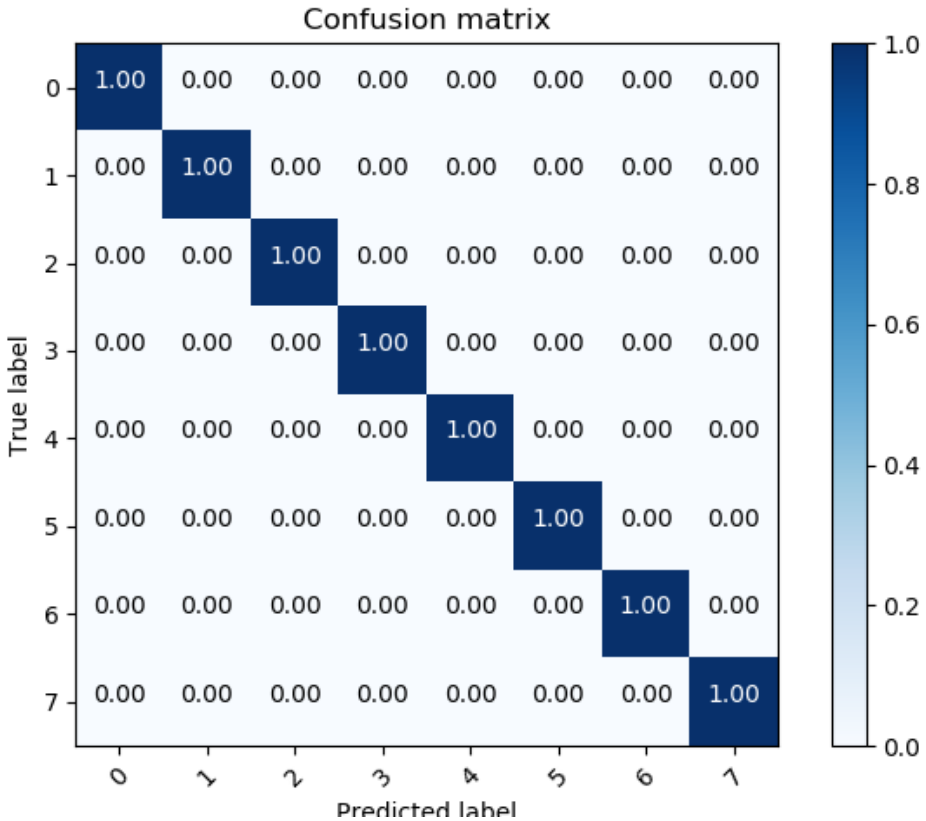

Figura 6.31: Matriu de confusió amb les dades d'entrenament del model Multi-Class SVM

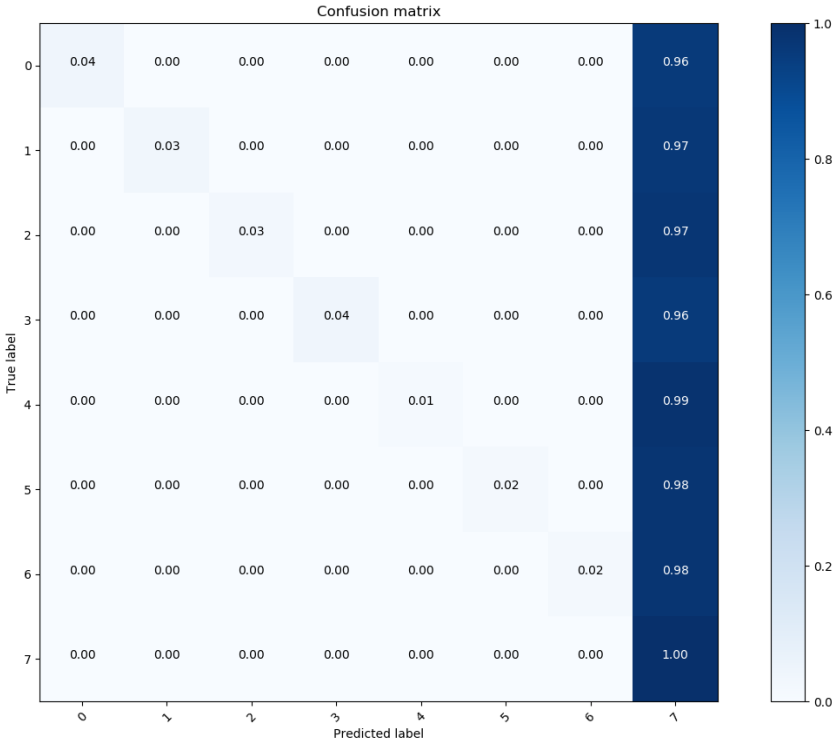

Figura 6.32: Matriu de confusió amb les dades de test del model Multi-Class SVM

A les figures 6.31 i 6.32 es mostren les matrius de confusió obtingudes de fer la rpedicció amb el conjunt d'entrenament i de test respectivament. Es destaca que en el cas de les dades de test, el model tendeix a predir la classe amb més observacions ja que pràcticament totes les observacions les ha predit com a 7 dies. Els resultats pel que fa a aquest model no són adequats per al problema que es vol resoldre.

6.6.2.5 Lasso (Regressió)

6.6.2.5.1 Descripció del mètode

Lasso [23] és un mètode de regressió. L'objectiu de Lasso és obtenir el subconjunt de variables que minimitza l'error de predicció per a una variable de resposta quantitativa. Lasso fa això imposant una restricció als paràmetres del model que provoquen que els coeficients de regressió d'algunes variables es redueixin cap a zero. Les variables amb un coeficient de regressió igual a zero després del procés de selecció són excloses del model. Les variables amb coeficients de regressió no nuls són més fortament associades amb la variable de resposta.

Els principal *Hiper-paràmetre* a és la constant de regularització (alpha). Quan alpha

= 0 aleshores el model és equivalent a la regressió lineal [24].

6.6.2.5.2 Execució i avaluació del mètode

Seguidament s'exposen els resultats d'entrenar i avaluar el model Lasso. El procediiment d'experimentació és similar a l'aplicat a la resta de models i s'explica a continuació:

1. Se separen les dades entre test i entrenament amb les mateixes proporcions que als casos anteriors.

2. S'executa l'entrenament utilitzant 20-fold CV amb el conjunt d'entrenament.

3. Es calculen els errors (mean squared error) tant de training com de test.

A la taula 6.11 es poden observar els valors de l'MSE (mean squared error) tant per les dades d'entrenament com de test. Vist que el rang de valors és de [0,7] els errors no semblen allunyar-se molt de l'objectiu. No obstant, si s'observa les figures 6.33 i 6.34 on es veu la gràfica valor predit vs valor real amb les dades d'entrenament i de test respectivament, es pot veure com no pocs casos s'allunyen molt del rang de valor desitjat. Aquest fet fa que aquest model no sigui interessant en un entorn de producció i, per tant, es descarta de la tria del model definitiu.

	MSE(Mean squared error)
Training	0.515
Test	0.521

Taula 6.11: MSE (training vs test) del model Lasso

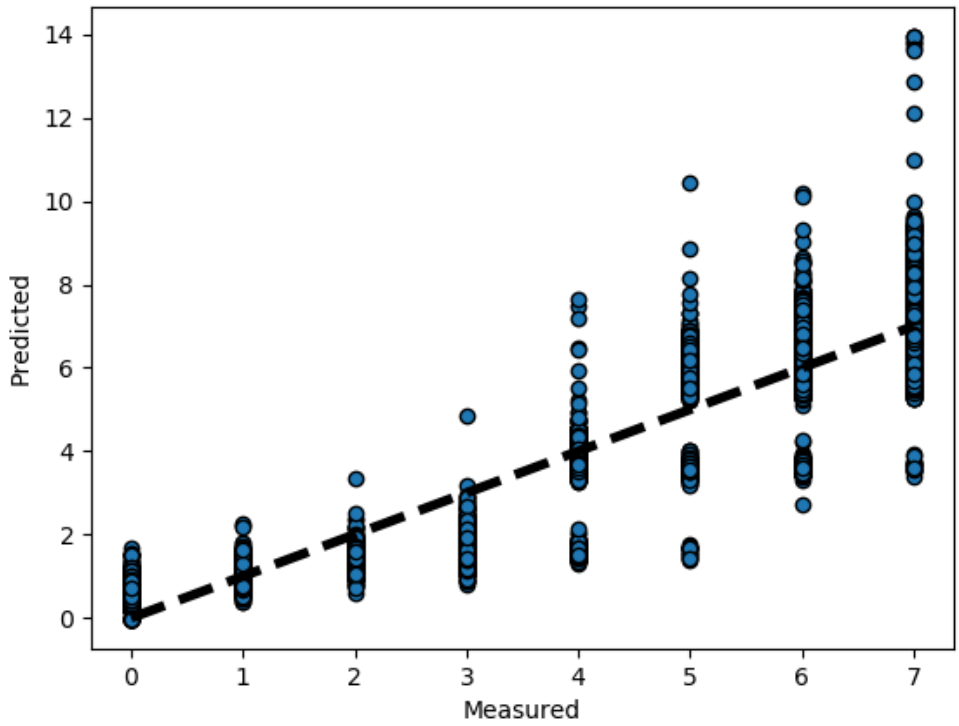

Figura 6.33: Gràfica de predicció del model Lasso amb les dades d'entrenament.

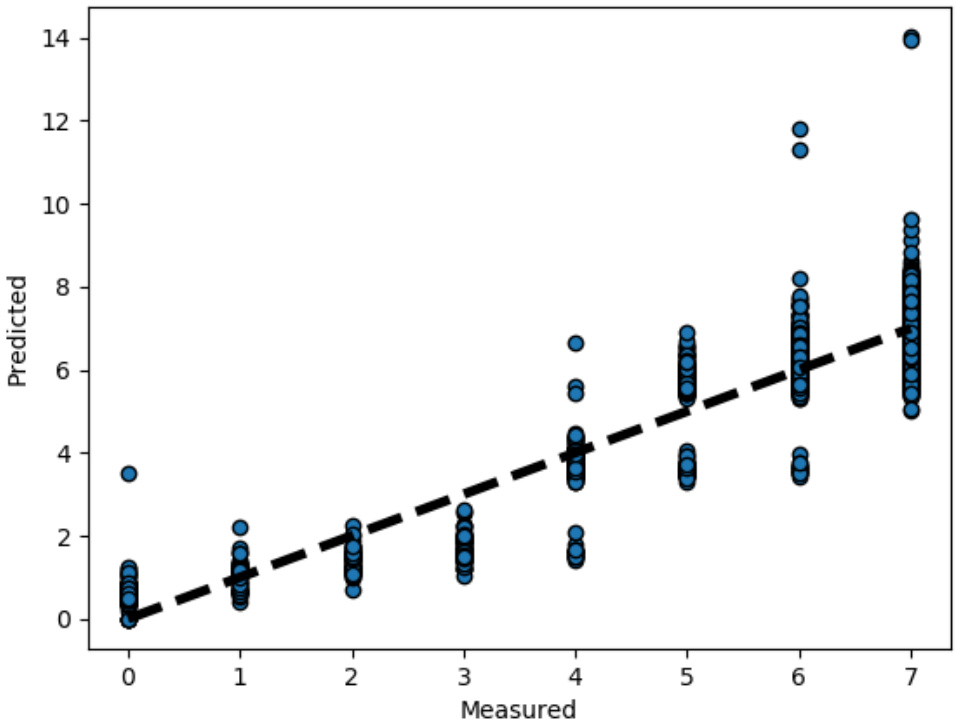

Figura 6.34: Gràfica de predicció del model Lasso amb les dades de test.

6.6.2.6 Selecció del model

Un cop entrenats i avaluats els 3 models: AdaBoost, RandomForest i Multi-Class SVM cal seleccionar el model que millor s'adapti al problema. A continuació es mostren les comparatives tant dels errors d'entrenament com de test:

	Encert d'entrenament (%)	Encert test (%)
AdaBoost	99.17	94.98
RandomForests	99.9	81.86
Multi-Class SVM	1.0	48.32

Taula 6.12: Comparativa dels encerts amb els conjunts d'entrenaments i de test dels diferents models analitzatts

A la taula 6.12 es pot observar la comparativa dels encerts dels diferents models. Com ja s'ha dit a la secció 6.6.2.4 el model Multi-Class SVM tendeix a l'overfitting molt més que els altres dos. Per tant, aquest es descarta degut a que els resultats són poc admissibles.

Ara bé els models AdaBoost i RandomForests són models que utilitzen algoritmes similars. En aquest cas els resultats són millors a priori pel cas d'AdaBoost. Aquesta diferència es podria deure a que els estimadors del model AdaBoost utilitzen totes les variables per predir el model mentre que en el cas del model RandomForests n'utilitzen un subconjunt. Per tant, i malgrat el model RandomForest pugui ser un model perfectament vàlid la selecció final que s'ha fet és el model **AdaBoost**.

6.6.3 Desplegament del model seleccionat

Un cop seleccionat el model, es necessita desplegar aquest model a la plataforma *Azure* per tal d'obtenir una APÌ REST que ens permeti cridar-la des de qualssevol aplicació. Als següents subapartats es descriuen els conceptes necessaris per entendre el procés de desplegament i també els passos seguits per a dur a terme el desplegament.

6.6.3.1 Conceptes importants pel desplegament

A continuació es descriuen els conceptes principals que cal conèixer per a dur a terme el procés de desplegament:

- **Azure ML Model Management**: el compte de gestió de models és un recurs Azure requerit per Azure ML Services per a la gestió de models. Es pot utilitzar per registrar models i fitxers de manifest, crear serveis web en contenidors i desplegar-los localment o al núvol.
- **Manifests**: quan el sistema de gestió de models implementa un model en producció, inclou un manifest que pot incloure el model, les dependències, les dades de mostra i l'esquema. El manifest és la recepta usada per crear una imatge del contenidor Docker. Amb la gestió de models, es poden generar automàticament els manifestos, crear versions diferents i gestionar aquests manifestos.
- **Imatges**: es poden utilitzar manifestos per generar (i regenerar) imatges Docker. Les imatges Docker creen flexibilitat per executar-les al núvol. Les imatges són autònomes i inclouen totes les dependències necessàries per fer prediccions amb dades noves als models.
- **Cotenidor Docker**: Defineix la estructura d'una imatge Docker. Un contenidor Docker permet exectuar ilimates imatges sobre la mateixa estructura.
- **Imatge Docker** [25]: una imatge del contenidor és un paquet executable lleuger, autònom, d'un programari que inclou tot el necessari per executar-lo: codi, temps d'execució, eines del sistema, biblioteques del sistema, configuració. Disponible tant per a aplicacions basades en Linux com per a Windows, el programari en contenidor sempre executarà el mateix, independentment del medi ambient. Els contenidors aïllen el programari del seu entorn, per exemple, les diferències entre entorns de desenvolupament i escenografia i ajuden a reduir els conflictes entre equips que executen diferents programes en la mateixa infraestructura.

- **Serveis**: La gestió de models permet implementar models com a serveis web. La lògica de servei web i les dependències s'engloben en una imatge. Cada servei web és un conjunt de contenidors basats en la imatge preparada per atendre sol·licituds a una URL determinada. Un servei web es compta com un desplegament únic.

6.6.3.2 Passos de desplegament

A continuació es descriuen passos seguits per a dur a terme el desplegament del model seleccionat utilitzant el servei *Azure ML Model management*. Els passos seguits són els que es descriuen a continuació:

1. Configurar un compte per utilitzar el servei *Azure ML Model management*. A [26] es descriuen els passos per a dur a terme aquesta tasca. Com a requisit cal tenir un compte de Microsoft Azure.

2. Guardar el model seleccionat un cop entrenat i validat. Per a fer-ho s'ha utilitzat la llibreria pickle [27] que permet serialitzar objectes python en fitxer. A continuació es mostra el codi desenvolupat.

```
import pickle
import sys
import os

#Train and evaluate model

...

#train model after parameter optimization

adaBoost.fit(X\_train, y\_train)

with open(os.path.join('.', 'outputs','AdaBoostModel.pkl'), 'wb') as f:
    pickle.dump(adaBoost, f)
```

3. Crear un fitxer d'esquema en format *.json*: es crear un fitxer d'esquema per tal de validar automàticament l'entrada i sortida del model del servei.

4. Crear un fitxer d'avaluació del model (Score.py): aquest fitxer serà executat pel servei cada vegada que es faci una crida. A cada crida el servei carregarrà i executar aquest fixer per tal de returnar les prediccions utilitzant el model. Aquest conté dues funcions:

   - Funció d'inicialització: té per objectiu carregar el fitxer que conté el model.

```
def init():
    global inputs\_dc, prediction\_dc
    from sklearn.externals import joblib
    import pickle

global model
# load the model file
#model = open('./outputs/AdaBoostModel.pkl', 'rb')
modelName = 'AdaBoostModel'
path = modelName +'.pkl'
model = joblib.load(path)
```

- Funció d'execució: funció que s'executa cada cop que es fa una crida al servei. Utilitza el model creat prèviament i retorna les prediccions.

```
def run(inputdf):
    import json
    pred = model.predict(inputdf)
    return json.dumps(str(pred[0]))
```

5. Registrar el model: un cop creats l'esquema i el fitxer d'avaluació, el següent pas és registrar el model al compte de gestió de models. Per a fer-ho s'ha utilitzat la següent comanda:

```
az ml model register --model .\AdaBoostModel.pkl --name AdaBoostModel.p
```

6. **Crear el manifest**: per a fer-ho s'ha utilitzat la següent comanda:

```
az ml manifest create --manifest-
name adaboostmodelmanifest -f
.\ScoreModel.py -r python -i adaBoostModel
-s .\service_schema.json
```

7. **Crear la imatge docker**: es crearà una imatge que contindrà el codi i les dependències. Per a fer-ho s'ha utilitzat la següent comanda:

```
az ml image create -n [image name] -manifest-id adaboostmodelmanifest
```

‘

8. **Crear i desplegar el servei**: un cop tenim la imatge creada i preparada es procedeix a crear i desplegar al núvol un servei amb la imatge creada i que contindrà una API Rest que permetrà executar aquest codi. Per a fer-ho s'ha utilitzat la següent comanda:

```
az ml service create realtime
--model-file .\AdaBoostModel.pkl -f
ScoreModel.py -n AdaBoostModelServices -s
service\_schema.json -r python -c  aml_config\conda_dependencies.yml
```

6.7 Desenvolupament de l'aplicació

Aquesta secció té com a finalitat explicar l'arquitectura i el desenvolupament de l'aplicació. L'objectiu és dotar d'una eina al pacient perquè aquest pugui veure quina serà l'evolució per un tractament determinat. Aquesta eina també conté elements de *Gamification* que tenen per objectiu motivar a l'usuari que faci ús d'aquesta eina.

S'ha decidit realitzar una **aplicació d'escriptori** multiusuari. Aquesta permetrà a l'usuari connectar-se amb unes credencials. Un cop connectat l'usuari podrà consultar la predicció del nombre de dies que haurà de dur màscara d'oxigen durant més de 3 hores per a les properes setmanes en forma de gràfica. A més a més aquesta aplicació ofereix un sistema de puntuació que cada usuari podrà comparar respecte la resta d'usuaris. Aquesta puntuació es podrà augmentar realitzant un seguit d'accions com ara utilitzar l'opció de consultar la predicció i també respondre un seguit de preguntes que poden ser d'ajuda als prescriptors.

6.7.1 Arquitectura

En aquest apartat es descriurà l'arquitectura utilitzada per al desenvolupament de l'aplicació. Com ja s'ha comentat anteriorment, aquesta aplicació és una aplicació d'escriptori. Aquesta aplicació s'encarrega de mostrar la informació a l'usuari i també de fer tot el procés de connexió amb la Base de dades així com demanar al Servei Machine Learning creat anteriorment les prediccions que l'usuari ha consultat.

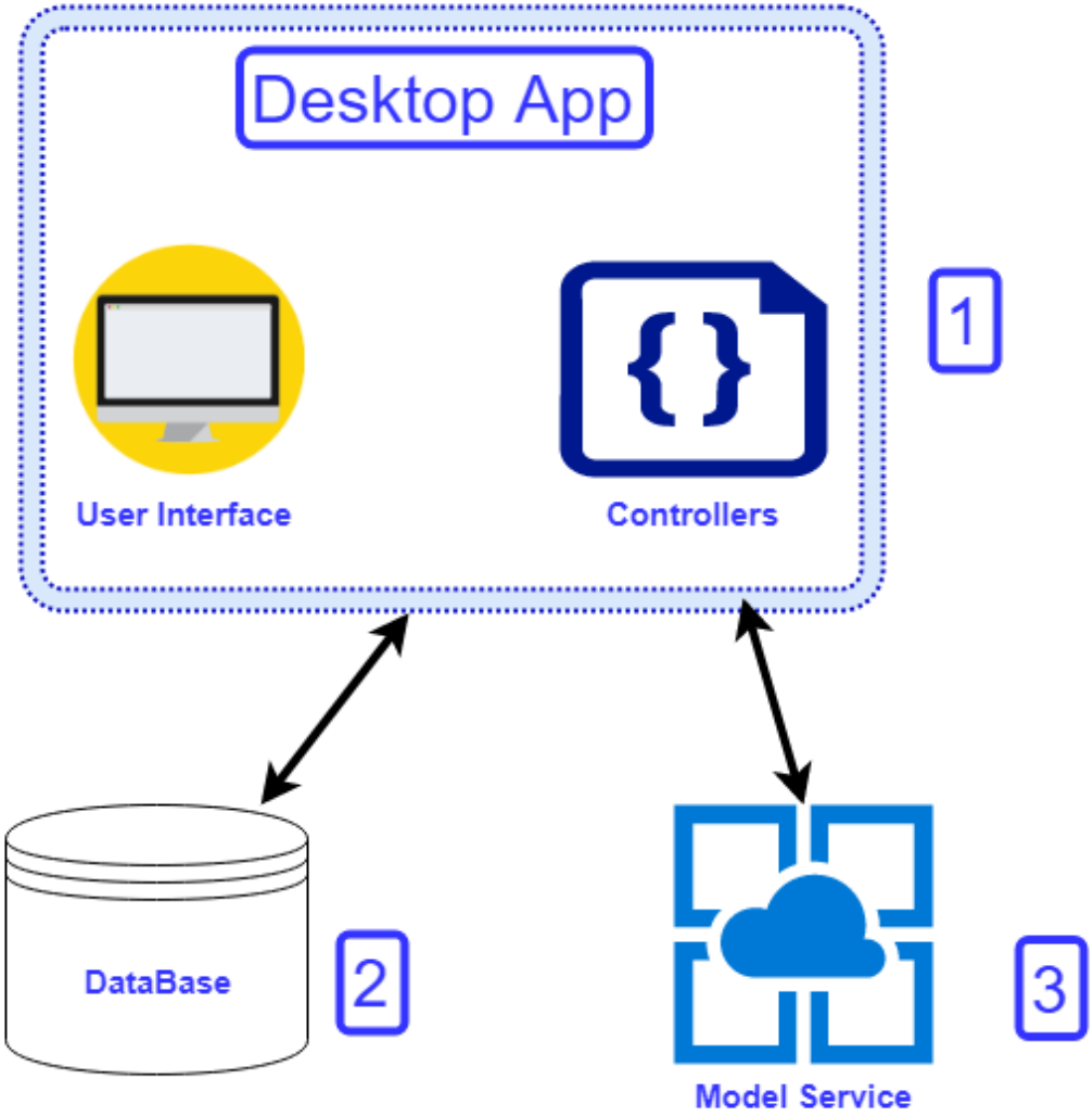

Figura 6.35: Esquema general de l'arquitectura de l'aplicació

Com es pot veure a la figura 6.35 l'aplicació consta del client d'escriptori(1) que interactua tant amb la base de dades (2) com amb el servei al cloud que allotja el model (3).

6.7.1.1 Base de dades

Un dels elements imprescindibles per a poder desenvolupar l'aplicació és la Base de Dades. Aquesta base de dades és la mateixa que s'ha utilitzat per desenvolupar el model (es pot consultar l'explicació de cadascuna de les taules a la secció 6.2.1.1). A més s'han creat dues taules addicionals: una per emmagatzemar la puntuació del rànquing del

pacient i l'altre per definir les qüestions i emmagatzemar les respostes dels usuaris. A continuació s'explica breument el contingut i la forma d'aquestes dues taules:

- **QuestionaireQuestions**: té per objectiu modelar les preguntes i respostes per usuari. Els camps rellevants són:
  - PatientId (enter): fa referencia a l'identificador del pacient que ha respost la pregunta.
  - QuestionText(string): es refereix al text de la pregunta.
  - Answer(string): fa referencia a la resposta del pacient.
- **UserRanquing**: té per objectiu modelar la puntuació d'un determinat pacient així com el seu nivell. Els camps més rellevants són:
  - PatientId (enter): fa referencia a l'identificador del pacient que ha respost la pregunta.
  - Puntuation (enter): puntuació de l'usuari.
  - Level (enter): nivell de l'usuari. Parteix de 1 quan l'usuari té entre 0 i 999 punts. Cada 1000 punts automàticament s'augmenta un nivell.

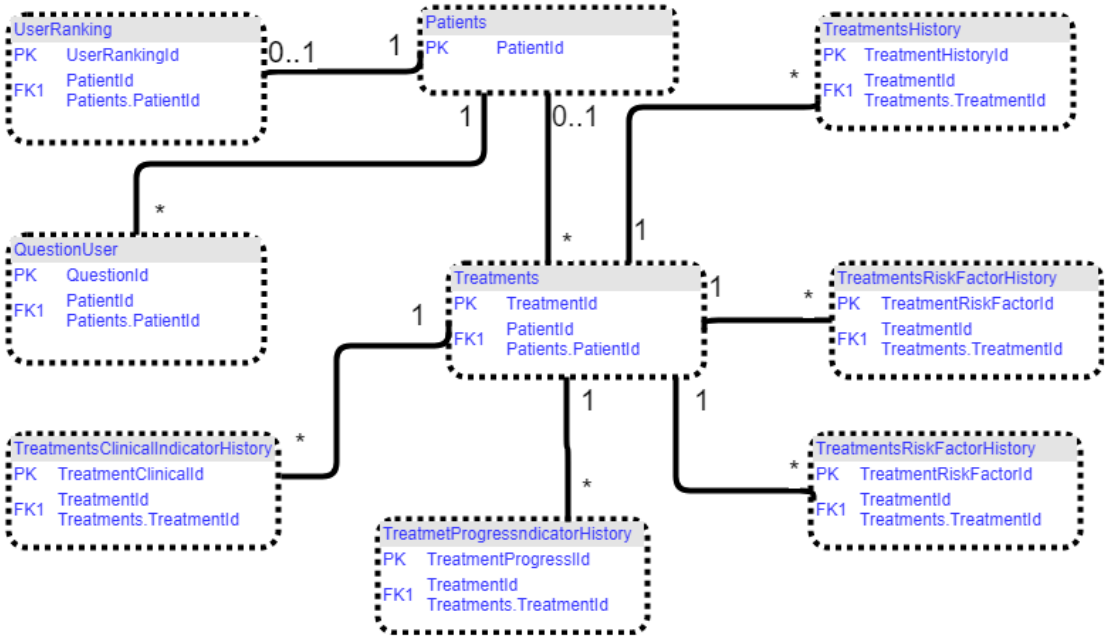

Figura 6.36: Esquema de les taules que s'usen a l'aplicació.

A la figura 6.36 es mostra l'esquema de les taules de la base de dades utilitzades per l'aplicació. Les taules referents als tractaments (Treatments, TreatmentsHistory, TreatmentsProgressIndicatorsHistory, TreatmentsClinicalIndicatorsHistory, TreatmentsRiskFactorHistory) i Patients s'utilitzen per recopilar les dades necessàries per a cridar al

servei Machine Learning i aquest pugui retornar les prediccions a partir d'aquestes dades. A més a més, la taula Patients juntament amb la taula UserRanking s'utilitzen per actualitzar i consultar les puntuacions el nivell tant de l'usuari com de la resta d'usuaris. Finalment, la taula Patients i QuestionUser s'utilitzen per mostrar les questions i actualitzar les respostes.

6.7.1.2 Client d'escritpori

El client d'escriptori té dues principals finalitats: construir un executable que permeti als usuaris finals interactuar i visualitzar de forma gràfica la informació que proporciona el sistema i per altra banda comunicar-se amb la base de dades i els serveis externs com el Model Service.

L'aplicació s'ha desenvolupat utilitzant la tecnologia .NET Framework [28] amb el llenguatge C#. NET Framework és una plataforma de codi obert desenvolupada per Microsoft que ofereix un seguit de funcionalitats bàsiques com llibreries d'accés al sistema operatiu, etc. El llenguatge C# és un llenguatge orientat a objectes de tipat fort.

La **interfície d'usuari** s'ha desenvolupat amb la tecnologia Windows Presentation Foundation (WPF) [29].WPF és un *framework* d'interfície d'usuari que permet crear aplicacions de client d'escriptori. La plataforma de desenvolupament de WPF admet un ampli conjunt de funcions de desenvolupament d'aplicacions, incloent un model d'aplicació, recursos, controls, gràfics, disseny, enllaç de dades, documents i seguretat. WPF utilitza el llenguatge de marcatge d'aplicacions extensible (XAML) per proporcionar un model declaratiu per a la creació de vistes. Les principals vistes de l'aplicació són:

- **Vista de Login** (LoginPage.xaml): permet a l'usuari introduir les seves credencials per accedir al contingut principal de l'aplicació.
- **Vista principal** (MainView.xaml): mostra informació bàsica de l'usuari (principalment el nom) així com la data. A més també ofereix un seguit de controls a l'usuari per consultar les prediccions per a la propera setmana i visualitzar aquestes en forma de gràfica.
- **Vista de rànquing** (RankingPage.xaml): mostra a l'usuari la puntuació actual que té així com una llista on es pot veure la puntuació ordenada de major a menor de les puntuacions de tots els usuaris del sistema.
- **Vista de qüestionaris** (QuestionariePage.xaml): mostra a l'usuari un seguit de qüestions i els controls necessaris per a respondre-les.

La vista principal ofereix un control per desconnectar-nos de l'aplicació i redirigeix a l'usuari a la vista de Login. Les vistes de rànquings i de qüestionaris ofereixen cadascun un control per redirigir a l'usuari a la vista principal.

Per altra banda, els **controladors** [8] s'encarreguen de recollir les accions que l'usuari ha realitzat a la vista i s'encarreguen de realitzar les accions sol·licitades i interactuar amb els sistemes externs com la Base de Dadaes i el servei ML. A l'aplicació existeixen dos tipus de controladors:

- **Controladors DB**: aquests controladors s'encarreguen de comunicar-se amb la base de dades. Els controladors són:
  - **PatientService**: s'encarrega d'interactuar amb la taula Patients de la Base de dades. La principal funció és consultar la informació principal de l'usuari i verificar que aquest existeix a l'hora d'entrar a l'aplicació.
  - **TreatmentService**: s'encarrega d'interactuar amb la taula Treatments. La principal funció és obtenir els tractaments associats a l'usuari.
  - **ClinicalIndicatorSerice**: interactua amb la taula TreatmentClinicalIndicatorHistory. S'encarrega d'obtenir els indicadors clínics de cadascun dels tipus sol·licitats associats a un tractament. Transforma els valors d'aquests de manera que siguin aptes com a entrada pel servei extern Model Service.
  - **ProgressIndicatorService**: interactua amb la taula TreatmentProgressIndicatorHistory. S'encarrega d'obtenir els indicadors de progrés de cadascun dels tipus sol·licitats associats a un tractament. Transforma els valors d'aquests de manera que siguin aptes com a entrada pel servei extern Model Service.
  - **ConsumptionService**: interactua amb la taula TreatmentConsumptionHistory. S'encarrega d'obtenir els valors de ratio de mesura associats a un tractament. Transforma els valors d'aquests de manera que siguin aptes com a entrada pel servei extern Model Service.
  - **RiskFactorService**: interactua amb la taula TreatmentRiskFactorHistory. S'encarrega d'obtenir els indicadors de risc de cadascun dels tipus sol·licitats associats a un tractament. Transforma els valors d'aquests de manera que siguin aptes com a entrada pel servei extern Model Service.
  - **UserRankingService**: interactua amb la taula UserRanking. Les seves funcions són obtenir les puntuacions dels usuaris i actualitzar aquestes cada vegada qeu un usuari faci alguna acció que impliqui augment de puntuació.
  - **QuestionService**: interactua amb la taula QuestionaireQuestions. La seva funció és obtenir el llistat de preguntes i guardar-ne la resposta de l'usuari.
- **Controlador Machine Learning**: existeix un controlador que té per objectiu consultar al servei extern al núvol (ML Service) les prediccions a partir de les dades obtingudes dels controladors anteriors així com de la vista principal (nombre de setmanes per les quals es volen les prediccions).

[8] Per a aquest document les paraules controlador i servei són sinònims. Quan es vol parlar de servei extern es diu explícitament

Finalment els **Models** són les representacins de la informació amb la qual el sistema opera, per tant gestionen tots els accessos a aquesta informació, tant consultes com actualitzacions. Envia a la 'vista' aquella part de la informació que en cada moment li sol·licita perquè sigui mostrada (típicament a un usuari). Les peticions d'accés o manipulació d'informació arriben al model a través del servei.

Per a crear els models s'ha utilitzat el "framework"**Entity Framework** [30]. Entity Framework (EF) és un *Object-relational mapper* (ORM) que permet construir automàticament classes C# a partir de taules d'una base de dades. El principal motiu d'ús d'aquest "framework"és que s'integra perfectament amb SQL Server [8] ja que ambdós tecnologies són propietàries de Microsoft. En aquest projecte s'ha utilitzat per construir els models que s'expliquen posteriorment.

A continuació es descriuen els models[9] i les seves principals característiques que inclou l'aplicació desenvolupada:

- **Patients**: representa la informació d'un pacient. Els principals atributs són:
  - Id: valor enter que representa l'identificador del pacient.
  - BirthDate: valor de tipus Date. Representa la data de naixement..
  - FullName: valor de tipus string. Representa el nom complet.
- **Treatments**: representa la informació d'un tractament. Els principals atributs són:
  - Id: valor enter que representa l'identificador del tractament.
  - PatientId: valor enter. Representa l'identificador del pacient associat al tractament.
- **ClinicalIndicators**: representa la informació d'un pacient. Els principals atributs són:
  - Id: valor enter que representa l'identificador de l'indicador clínic.
  - CreationDate: valor de tipus Date. Representa la data de creació de l'indicador.
  - Value: valor de tipus double. Indica el valor numèric de l'indicador.
  - ClinicalTypeId: valor de tipus enter. Indica el tipus d'indicador clínic.
- **RiskFactorIndicators**: representa la informació d'un pacient. Els principals atributs són:
  - Id: valor enter que representa l'identificador de l'indicador de risc.

[9] Els models descrits poden tenir més característiques que no s'expliquen en detall ja que no són utilitzades per l'aplicació ni tenen especial rellevància.

  - CreationDate: valor de tipus Date. Representa la data de creació de l'indicador.
  - Value: valor de tipus double. Indica el valor numèric de l'indicador.
  - RiscFactorType: valor de tipus enter. Indica el tipus d'indicador de risc.

- **ProgressIndicators**: representa la informació d'un pacient. Els principals atributs són:
  - Id: valor enter que representa l'identificador de l'indicador de progrés.
  - CreationDate: valor de tipus Date. Representa la data de creació de l'indicador.
  - Value: valor de tipus double. Indica el valor numèric de l'indicador.
  - ClinicalTypeId: valor de tipus enter. Indica el tipus d'indicador de progrés.

- **UserRanking**: representa la informació d'un pacient. Els principals atributs són:
  - PatientId: enter. Representa l'identificador del pacient associat al rànquing.
  - Punctuation: enter. Representa la puntuació actual del pacient.
  - UserLevel: enter. Representa el nivell actual de l'usuari. Aquest nivell es calcula a partir de la puntuació. Cada 1000 punts s'augmenta un nivell.

- **Question**: representa la informació d'un pacient. Els principals atributs són:
  - Id: enter. identificador de la qüestió.
  - DefaultText: string. Identifica el text de la pregunta.

- **ModelEntity**: representa els atributs que necessita el model per a fer la predicció. Aquest atributs són:
  - PatientAge: enter. Representa l'edat del pacient.
  - PatientId: enter. Representa l'identificador del pacient.
  - TreatmentId. enter. Representa l'identifiacdor del tractament associat al pacient.
  - WeekCreated. enter. Representa el nombre de setmana (a partir del 17/6/2009).
  - ClinicalValue5: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 5.
  - ClinicalValue7: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 7.
  - ClinicalValue8: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 8.
  - ClinicalValue9: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 9.

- ClinicalValue13: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 13.
- ClinicalValue15: double. Representa la mitjana setmanal de tots els dels valors dels indicadors clínics de tipus 15.
- ProgressValue2: double. Representa la mitjana setmanal de tots els dels valors dels indicadors de progres de tipus 2.
- ProgressValue4: double. Representa la mitjana setmanal de tots els dels valors dels indicadors de progrés de tipus 4.
- Consumption: double. representa la mitjana setmanal dels valors consum associats al tractament.

6.7.2 Elements de Gamification

Com ja s'ha dit a la secció 1.2 un dels objectius del projecte és introduir elements de **Ludificació** o *Gamification* en anglès. Aquest objectiu té com a finalitat motivar que els usuaris utilitzin l'aplicació proposant-los reptes i premiant-los per fer ús d'aquesta.

En aquesta aplicació s'ha implementat principalment un sistema de puntuació i nivells que premia a l'usuari per realitzar diferents accions. El sistema de puntuació funciona de la següent manera:

- Tot usuari comença inicialment amb 0 punts i es situa al nivell 1.
- Cada vegada que l'usuari consulti les prediccions se li sumaran 5 punts a la seva puntuació.
- Cada vegada que l'usuari respongui una pregunta del qüestionari se li sumaran 10 punts.
- Cada vegada que l'usuari sumi 1000 punts, s'augmentarà un nivell (nivell 2 pels 1000 punts, nivell 3 pels 2000, etc.)

Aquest sistema de puntuació premia més el fet de respondre una pregunta ja que és una acció que requereix (o hauria de requerir) un esforç de pensar com respondre bé la pregunta. A més a més, aquesta informació ajuda també als prescriptors. S'ha considerat per tant que aporta més valor respondre una qüestió.

L'usuari també podrà consultar el rànquing de puntuació i nivell de tot el conjunt d'usuaris. Això és així ja que s'incita a la competència entre els usuaris.

A partir d'aquestes puntuacions es podrien premiar de forma directa als usuaris amb premis "físics"a partir de les puntuacions. Per exemple, es podria regalar un viatge o algun producte tecnològic a l'usuari amb més puntuació personal o qualssevol regal o també donar algun premi quan s'arribi a un determinat nivell etc. No obstant, això només són propostes en el cas que algun sistema derivat del que es presenta en aquest treball es desplegui en un entorn real i queda fora de l'abast d'aquest projecte.

6.7.3 Funcionalitats

A continuació es descriuen les principals funcionalitats que integra l'aplicació. Aquesta aplicació està destinada a **pacients** que tenen un historial mèdic a la Base de Dades. El pacient podrà accedir a l'aplicació mitjançant l'identificador d'usuari. Un cop introduït i que aquest sigui vàlid, l'usuari veurà una finestra principal on podrà consultar les prediccions per les properes setmanes així com accedir a la finestra per respondre els qüestionaris o veure el rànquing de puntuacions.

6.7.3.1 Inici de sessió

Quan s'inicia l'aplicació es mostra una finestra que permet l'accés a les principals funcionalitats a través de d'un formulari d'identificació.

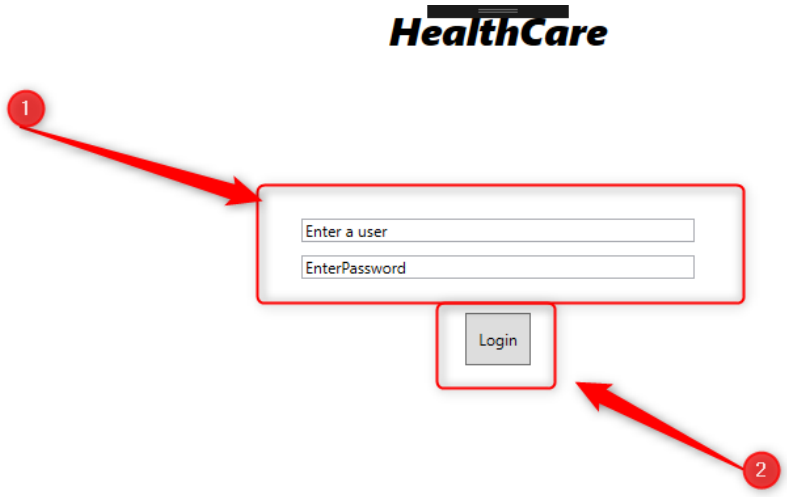

Figura 6.37: Finestra d'entrada a l'aplicació

A la figura 6.38 es pot veure com és la finestra principal d'entrada a l'aplicació. Aquesta conté els següents elements:

1. (1): Formulari d'accés: l'usuari ha d'Introduir NOMÉS l'identificador d'usuari. El camp Password només té una funció estètica per a aquesta aplicació. Si l'usuari no entra un identificador correcte, es mostrarà un text com el de la següent figura:

2. (2): Si s'ha introduit un identificador correcte, l'aplicació mostrarà la finestra principal de l'aplicació, altrament es mostrarà un text com el de la següent figura:

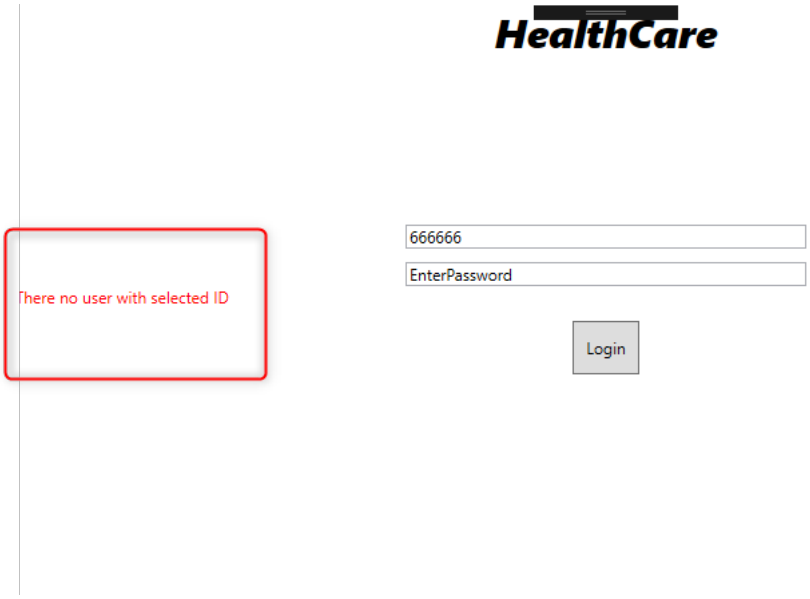

Figura 6.38: Finestra d'inici de sessió quan l'Identificador no és vàlid

Cal remarcar que aquesta funcionalitat s'ha implementat per facilitar les proves i que en un entorn real s'implementaria completament de forma similars a les que implementen la majoria d'aplicacions multiusuari.

6.7.3.2 Consultar prediccions

Pel que fa a la pestanya de **prediccions** es podran consultar les prediccions segons un tractament determinat. A la figura 6.39 es pot veure el contingut de la pestanya de prediccions que es descriu a continuació: A la figura 6.39 es mostra l'estat de l'aplicació un cop l'usuari ha introduït l'identificador. A continuació es descriuen els diferents elements:

- (1): Selector que permet seleccionar un dels diferents tractaments de l'usuari.
- (2): Camp que permet seleccionar el nombre de setmanes per les quals es vol la predicció. (màxim de 5).
- (3): botó per demanar les prediccions. Només s'executarà si el camp (9) té un valor vàlid.
- (4): gràfic on es mostren les prediccions.

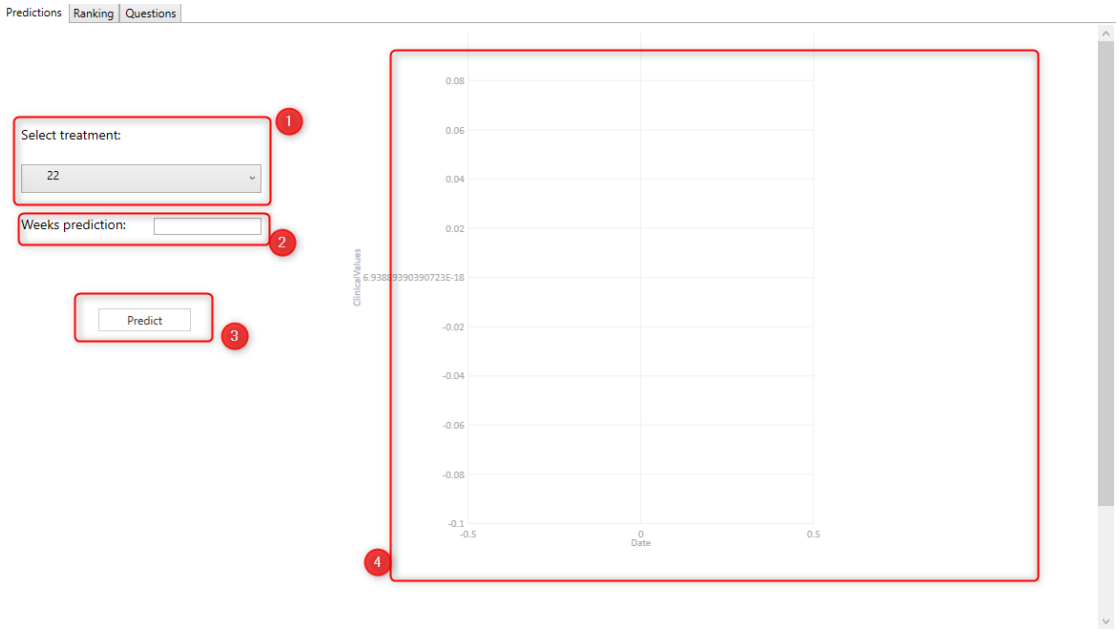

Figura 6.39: Finestra principal de l'aplicació

Per poder veure en el gràfic les prediccions cal seleccionar el tractament desitjat, descriure el nombre de setmanes per les quals es vol la predicció (màxim de 5) i fer clic al botó (10 en la imatge). Si tot és correcte es mostrarà un gràfic com el de la figura 6.40. Al gràfic es pot veure el valor de la predicció per les següents X setmanes desitjades:

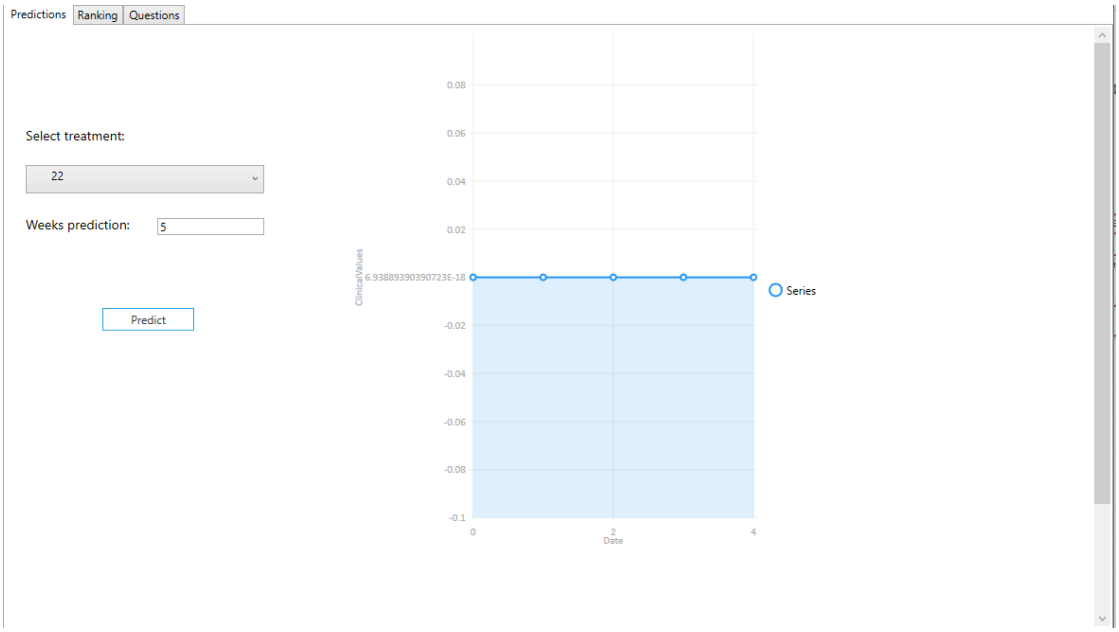

Figura 6.40: Exemple de gràfica de prediccions futures

A més a més, executar aquesta acció aporta un valor ja que permet al sistema recollir informació.[10]. Es mostrarà una finestra temporal amb la puntuació obtinguda com la de la imatge 6.41.

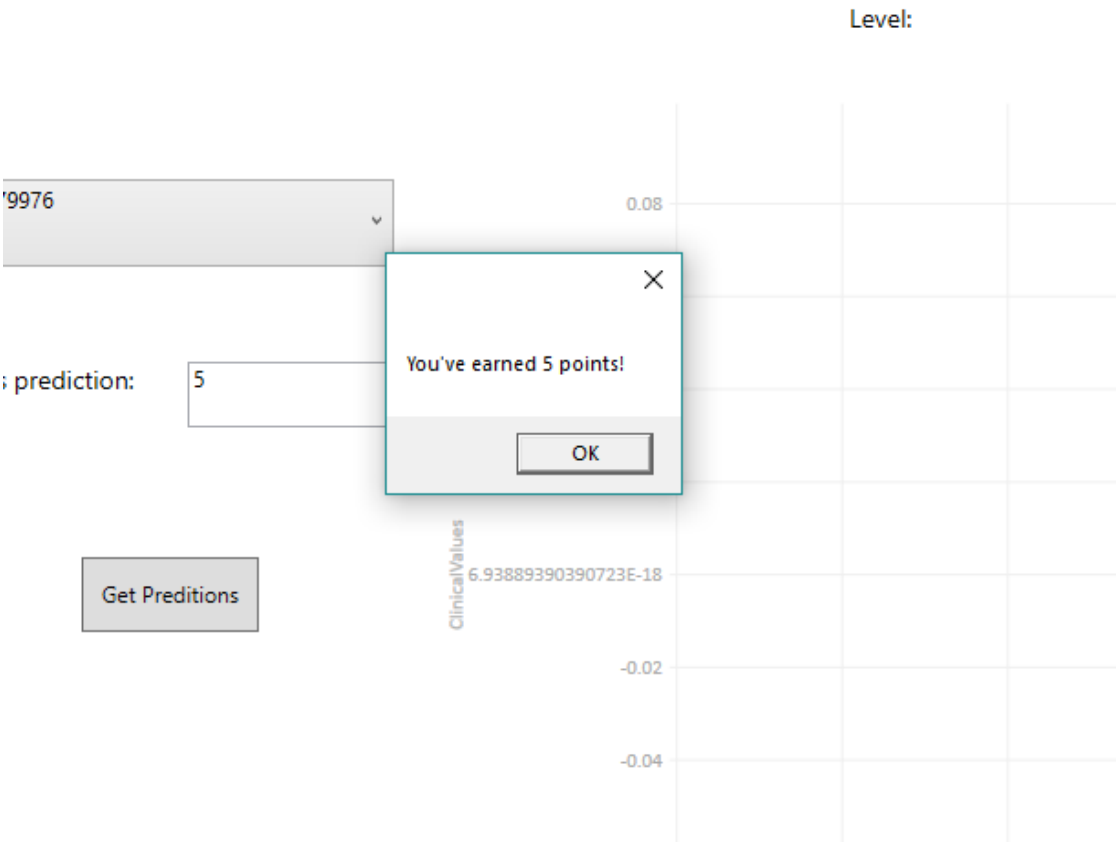

Figura 6.41: Finestra "pop up"de puntuació després de demanar prediccions.

6.7.3.3 Visualitzar la puntuació i el rànquing

Aquesta funcionalitat permet a l'usuari visualitzar la puntuació de la resta d'usuaris per poder comparar la seva amb la resta. La finalitat d'aquesta funcionalitat és incitar a l'usuari a utilitzar l'aplicació de manera que comparant-se amb els altres l'inciti a millorar la seva posició en el rànquing.
La figura 6.42 mostra la finestra principal on es mostra el rànquing. A continuació es descriuen els principals elements:

- (1): tornar enrere a la finestra principal de l'aplicació.

[10]En aquesta versió del projecte no s'ha recopilat informació ni s'ha fet cap tractament amb les prediccions. L'objectiu és premiar a l'usuari per realitzar accions que aportin valor

- (2): Taula de visualització del rànquing. Es mostra la posició l'identificador de l'usuari, la puntuació i el nivell per a cada usuari ordenats de major a menor puntuació.
- (3), (4), (5), (6) i (7): botons de navegació. A (7) escollim el nombre d'elements a buscar(10,20 o 30). (3) i (6) ens permeten cercar els primers usuaris i els últims. (4) i (5) accedeixen als anteriors i següents elements respectivament.

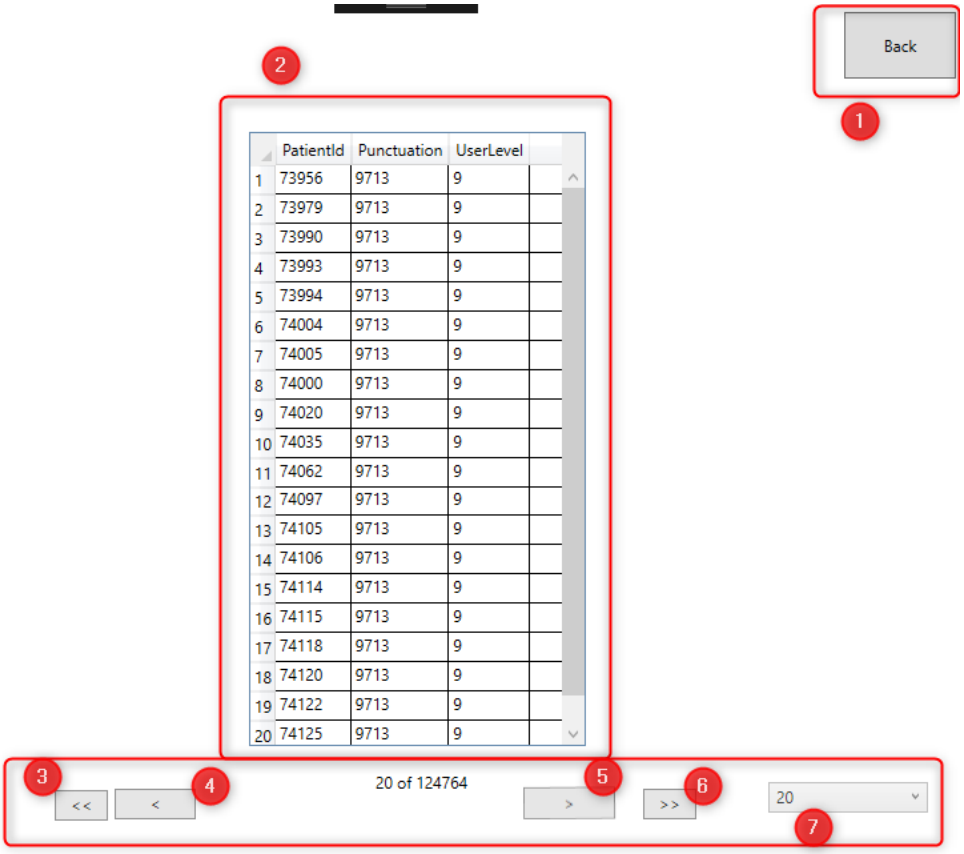

Figura 6.42: Finestra de visualització del rànquing.

6.7.3.4 Respondre qüestionaris

Una altra funcionalitat que aporta l'aplicació és la de respondre qüestionaris. La finalitat d'aquesta és incitar a l'usuari a respondre preguntes que podrien ser d'utilitat per a millorar tant el model de dades, com la informació que tenen els prescriptors sobre l'estat del pacient o el coneixement mèdic que té cada pacient sobre el seu tractament. S'ha de deixar clar, que no hi ha cap implementació per recopilar les respostes de l'usuari. L'objectiu és mostrar un element més de Ludificació que pot ser efectiu i no pas implementar la recollida i anàlisi de les respostes.

La figura 6.43 mostra la finestra principal del qüestionari. A continuació es descriuen els principals elements:

- (1): Pregunta: per cada pregunta es mostra la pregunta així com un camp de text per respondre i un botó per salvar aquesta. Quan es salvi aquesta resposta, l'usuari obtindrà 10 punts que se sumaran a la seva puntuació total. Es mostrara una finestra com la figura 6.44

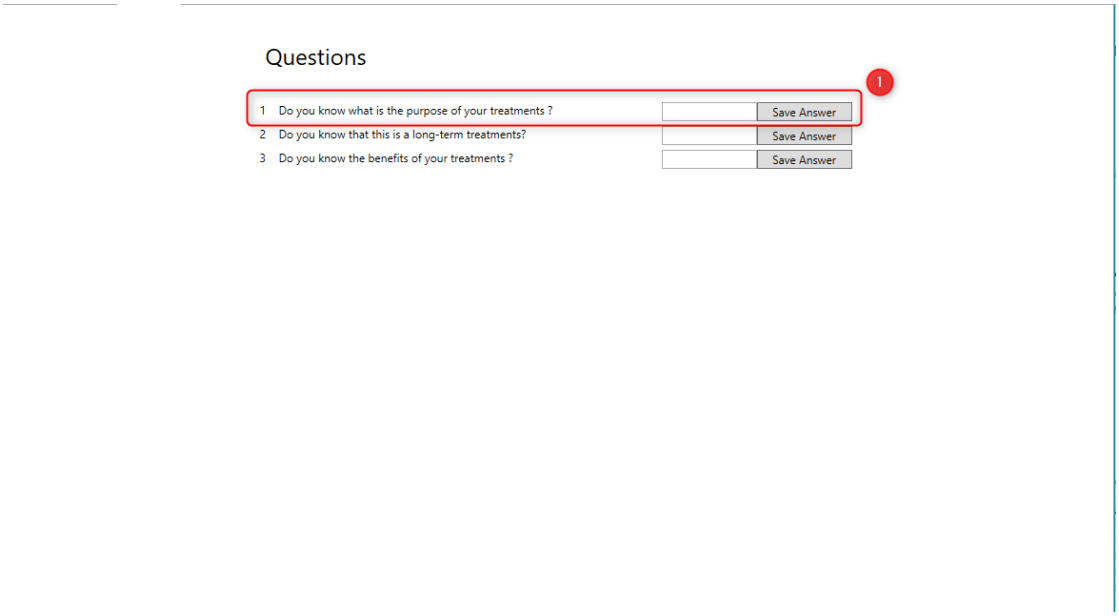

Figura 6.43: Finestra de visualització del qüestionari.

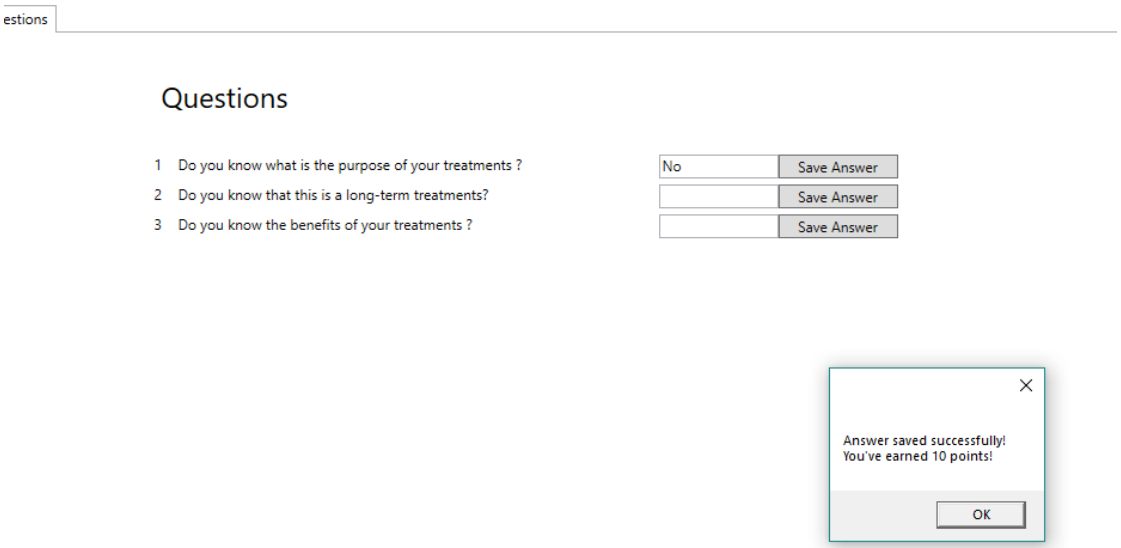

Figura 6.44: Missatge de puntuació obtinguda després de salvar la resposta d'una pregunta.

7 Conclusions i futures extensions

7.1 Conclusions tècniques

En aquesta secció es resumeixen totes les fases del projecte i s'exposen les conclusions extretes un cop finalitzat el desenvolupament d'aquest.

El projecte consisteix a crear un entorn o plataforma de suport a pacients amb malalties respiratòries que té per objectiu millorar la qualitat de vida o obrir una possible porta a que això sigui possible mitjançant futures extensions. Concretament, s'ha creat una plataforma que s'ajuda de tècniques de **Machine Learning** que consumeixen les dades disponibles dels pacients per tal d'intentar predir la futura evolució de diferents aspectes relacionats amb els tractaments associats a cada pacient. Per a aquest projecte, s'ha decidit predir el nombre de dies setmanals que cada pacient ha d'utilitzar la màscara d'oxigen (a tots els tractaments seleccionats l'usuari utilitza màscares d'oxigen). Aquest indicador és important, ja que mesura la qualitat de vida del pacient degut a que portar la màscara d'oxigen implica limitacions en la mobilitat. A més a més, també s'han introduït tècniques de **Ludificació** (Gamification en anglès) per tal de motivar a l'usuari a utilitzar la plataforma desenvolupada.

El primer pas desenvolupat ha estat obtenir la font d'informació necessària per tal de poder desenvolupar i alimentar models predictius. Aquestes dades provenen d'una base de dades relacional cedida pel client. Aquestes dades estan totalment anonimitzades per tal de de complir amb la legislació vigent. Un cop obtinguda la font d'informació, s'han explorat aquestes dades i adaptat per tal de poder desenvolupar els models predictius.

Un cop obtinguda la font d'informació i adaptades les dades, s'ha dut a terme el preprocessament de les dades i selecció de variables més importants. Seguidament, s'ha abordat el dubte de plantejar el problema com a classificació o bé regressió. S'han experimentat les dues opcions i s'ha arribat a la conclusió que la **classificació** s'adapta millor al problema i a la plataforma en general. Decidit doncs el tipus de problema, s'ha experimentat amb diferents algoritmes o mètodes de classificació s'ha arribat a la conclusió que el mètode **AdaBoost** és el que aporta uns resultats més adequats per a dur a terme les prediccions.

Un cop escollit el model el següent pas ha estat desplegar aquest al núvol per tal de poder utilitzar-lo en qualssevol plataforma. S'ha utilitzat la plataforma Azure per tal d'allotjar un servei Web que permeti a qualssevol aplicació o plataforma fer ús d'ell per tal de demanar prediccions aportant les dades necessàries. Aquest desplegament s'ha dut a terme amb la plataforma Azure Machine Learning Services integrada en l'ecosistema

d'Azure. Aquesta plataforma ha aportat eines molt útils per tal de poder dur a terme aquest desplegament.

Finalment s'ha desenvolupat una aplicació d'escriptori que fa ús d'aquest servei Web creat i també s'hi han implementat tècniques bàsiques de Ludificació. Aquestes tècniques consisteixen a premiar a l'usuari per a realitzar diferents accions com ara consultar les prediccions o bé respondre petits qüestionaris que podrien servir per obtenir informació extra i millorar el model.

Finalment i com a conclusió global del projecte, el més destacat és que s'ha estat capaç de partir de zero amb unes dades que no estaven adaptades per a resoldre el problema i arribar a construir una plataforma amb la qual l'usuari pot experimentar.
Cal deixar clar que es tracta d'una prova de concepte que no pretén ser una eina apte ja per entrar en producció. No obstant, l'objectiu principal, que és demostrar que és possible utilitzar tècniques relacionades amb l'anàlisi de dades i que aquestes puguin realment ser d'interès a la pràctica dels usuaris s'ha assolit.

7.2 Conclusions personals

Personalment cal dir que tenir l'oportunitat de treballar amb tècniques que estan en auge i que seran molt importants en un futur i al present com són la intel·ligència Artificial i el *Machine Learning* ha estat una una experiència d'aprenentatge molt intensa i alhora satisfactòria.
S'ha de destacar no obstant, que la tasca d'enfrontar-se sol al problema degut a que el client tenia poc o cap coneixement sobre com solucionar el problema a abordar i la inexperiència professional de l'autor d'aquesta memòria i desenvolupador únic del projecte, no ha estat gens fàcil i una experiència segurament necessària de cara a l'entorn professional. Aquest fet ha provocat la necessitat de buscar les solucions al problema sense l'ajuda d'experts. Aquest fet, no obstant també ha estat una experiència gratificant ja que ha permès oferir un determinat tipus de solucions i obrir un camp de possibles solucions a problemes que fins i tot ni es plantejaven solucionar.

Un altre aspecte que es destaca és el fet de tenir l'oportunitat de millorar la qualitat de vida de les persones. Malgrat ser una prova de concepte i que, probablement la millora sigui petita. Treballar per ajudar a millorar la qualitat de vida de les persones és una experiència que es valora molt i aporta sentit a treballar, no només com a enginyer, sinó en general com treballador.

Per concloure, en general l'experiència de realitzar un projecte des de l'inici, plantejarlo, afrontar els problemes sorgits, buscar-ne solucions és una experiència intensa alhora que interessant. Per altra banda, treballar amb tecnologies relacionades amb el món de l'anàlisi i experimentació amb les dades però sense la guia de cap expert també ha estat una experiència que de ben segur servirà de cara al futur professional.

7.3 Futures extensions

En aquesta secció s'exposen les futures extensions i/o millores les quals es podrien plantejar i implementar en un futur.
La primera extensió que es podria plantejar i implementar és la d'utilitzar més dades dels pacients i explorar encara més la base de dades original per tal d'obtenir models predictius més precisos i exactes millorant així la qualitat de les prediccions.

Per altra banda, es podrien crear diferents models que fessin prediccions de diferents variables relacionades amb els tractaments com algun indicador de risc o bé de progrés dels pacients. Aquests models es podrien allotjar en diferents serveis al núvol dins la plataforma Azure. Alhora es podrien explorar els diferents serveis al núvol que existeixen per emmagatzemar serveis i aplicacions i seleccionar el que millor s'adaptés a les necessitats o possibilitats.

Una possible extensió del projecte seria utilitzar les dades que es van recopilant dels pacients i utilitzar aquestes per aplicar tècniques de re-entrenament del model per adaptar les prediccions a les noves dades recopilades. L'eina Azure Machine Learning Services [31] ofereix eines per a dur a terme aquesta tasca de forma automàtica.

A més a més, per implementar aquesta eina en producció seria necessari i interessant implementar un mètode d'autenticació que permetes als usuaris connectar-se a l'aplicació de forma remota. També es podria migrar l'aplicació a diferents tipus de plataformes ja siguin Web o mòbil ja que són tecnologies que permetrien utilitzar l'aplicació des de qualsevol dispositiu connectat.

Bibliografia

[1] MicrosoftAzure, "Azure machine learning services schema." https://docs.microsoft.com/en-us/azure/machine-learning/preview/media/overview-what-is-azure-ml/aml-concepts.png.

[2] *Ley 41/2002 bàsica reguladora de la autonomia del paciente y de, 2002.*, vol. 41. 2002 ed., 2002.

[3] R. Pradeebha, "Prediction of lung disease using hog features and machine learning algorithms," Jan 2016. http://www.ijircce.com/upload/2016/january/13*Prediction.pdf*.

[4] Wikipedia, "Ludificació," Mar 2018. https://ca.wikipedia.org/wiki/Ludificaci\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{o\global\mathchardef\accent@spacefactor\spacefactor}\accent19o\egroup\spacefactor\accent@spacefactor.

[5] H. Suresh, N. Hunt, A. E. W. Johnson, L. A. Celi, P. Szolovits, and M. Ghassemi, "Clinical intervention prediction and understanding using deep networks," *CoRR*, vol. abs/1705.08498, 2017.

[6] X. Wang, D. Sontag, and F. Wang, "Unsupervised learning of disease progression models," 08 2014.

[7] Microsoft, "Team foundation server." https://www.visualstudio.com/tfs/p=DevEx,5068.1.

[8] Microsoft, "Run sql server on your favorite platform." https://www.microsoft.com/en-us/sql-server/sql-server-2017.

[9] P. S. Foundation, "Python." https://www.python.org/.

[10] P. project, "Python data analysis library¶." https://pandas.pydata.org/.

[11] "Pyodbc." http://mkleehammer.github.io/pyodbc/.

[12] Wikipedia, "Feature selection." https://en.wikipedia.org/wiki/Feature_selection.

[13] A. Backaria, "Recursive feature elimination with scikit learn." https://medium.com/@aneesha/recursive-feature-elimination-with-scikit-learn-3a2cbdf23fb7.

[14] Microsoft, "Microsoft azure." https://azure.microsoft.com/en-us/overview/what-is-azure/https://azure.microsoft.com/en-us/overview/what-is-azure/.

[15] Microsoft, "Microsoft azure machine learning services." https://docs.microsoft.com/en-us/azure/machine-learning/preview/.

[16] Microsoft, "Microsoft azure machine learning workbench." https://docs.microsoft.com/en-us/azure/machine-learning/preview/quickstart-installation.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] Wikipedia, "Adaboost." https://en.wikipedia.org/wiki/AdaBoost.

[19] Wikipedia, "Boosting (machine learning)." https://en.wikipedia.org/wiki/Boosting_(machine_learning).

[20] Wikipedia, "Decision trees." https://en.wikipedia.org/wiki/Decision_tree).

[21] WikiPedia, "Random forest." https://en.wikipedia.org/wiki/Random_forest.

[22] Wikipedia, "Support vector machine." https://en.wikipedia.org/wiki/Support_vector_machine.

[23] Wikipedia, "Lasso." https://en.wikipedia.org/wiki/Lasso_(statistics).

[24] Wikipedia, "Linear regression." https://en.wikipedia.org/wiki/Linear_regression.

[25] Docker, "Docker containers." https://www.docker.com/what-container.

[26] M. Azure, "Model management setup." https://docs.microsoft.com/en-us/azure/machine-learning/preview/model-management-configuration.

[27] Python, "Pickle: python object serialization." hhttps://docs.python.org/3/library/pickle.html.

[28] Microsoft, "What is .net?." https://www.microsoft.com/net/learn/what-is-dotnet.

[29] Microsoft, "Getting started with wpf." https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf.

[30] Microsoft, "Entity framework." https://docs.microsoft.com/en-us/ef/.

[31] Azure, "Retraining and updating azure machine learning models with azure data factory." https://azure.microsoft.com/es-es/blog/retraining-and-updating-azure-machine-learning-models-with-azure-data-factory/.